CMR INSTITUTE OF TECHNOLOGY

(Approved By AICTE-New Delhi, Affiliated to JNTUH)

Kandlakoya(V), Medchal Road, Hyderabad-501401

Department of CSE(AI & ML)

LAB MANUAL:

Name of the Lab: Robotic Process Automation Lab (22CSPC65)

Branch : CSE(AI&ML)

Year : III Year II semester

Regulation : R22

A.Y. : 2024-2025

Faculty Name: C.B. David Joel Kishore

Vision: Develop competent software professionals, researchers and entrepreneurs to serve global society.

Mission: The department of **Computer Science and Engineering** is committed to

- create technocrats with proficiency in design and code for software development
- adapt contemporary technologies by lifelong learning and face challenges in IT and ITES sectors
- quench the thirst of knowledge in higher education, employment, R&D and entrepreneurship

I. Programme Educational Objectives (PEOs): Engineering Graduates will

1. Pursue successful professional career in IT and IT-enabled sectors.
2. Pursue lifelong learning skills to solve complex problems through multidisciplinary-research.
3. Exhibits professionalism, ethics and inter-personal skills to develop leadership qualities.

II. Programme Outcomes (POs): Engineering Graduates will be able to

1. Apply mathematics, science, engineering fundamentals to solve complex engineering problems.
2. Identify, formulate and analyze complex engineering problems to reach substantiated conclusions.
3. Design and develop a component/system/process to solve complex societal engineering problems.
4. Design and conduct experiments to analyze, interpret and synthesize data for valid conclusions.
5. Create, select and apply modern tools, skills, resources to solve complex engineering problems.
6. Apply contextual engineering knowledge to solve societal issues.
7. Adapt modern engineering practices with environmental safety and sustainable development.
8. Apply professional code of ethics, responsibilities and norms in engineering practices.
9. Compete as an individual and/or as a leader in collaborative cross cultural teams.
10. Communicate effectively through technical reports, designs, documentations and presentations.
11. Endorse cognitive management skills to prepare project report using modern tools and finance.
12. Engage in independent and life-long learning in the broad context of technological changes.

III. Programme Specific Outcomes (PSOs): Engineering Graduates will be able to

1. Design and develop Computer-Based-Systems using Algorithms, Networks, Security, Gaming, Full Stack, DevOps, IoT, Cloud, Data Science and AI&ML.
2. Apply data analytics to solve real world problems.

System Requirements:

	Minimum	Recommended
CPU Cores	2 x 1.8 GHz 32-bit (x86)	4 x 2.4 GHz 64-bit (x64)
RAM	4 GB of usable system memory	8 GB
Disk Space	3.5 GB for new installations, 5 GB for upgrades (including temporary files required during installation)	N/A

Lab Objectives:

Upon successful completion of this Lab the student will be able to:

1. Install RPA packages
2. Apply variables, data types, control statements in designing RPA
3. Make use of data manipulation, recording and scrapping techniques
4. Use selectors, data tables in excel for automation
5. Develop email and PDF automation

SYLLABUS

ROBOTIC PROCESS AUTOMATION LAB

Course	**B.Tech.-VI-Sem.**	**L**	**T**	**P**	**C**
Course Code	**22CSPC65**	**-**	**-**	**2**	**1**

Course Outcomes (COs) & CO-PO Mapping (3-Strong; 2-Medium; 1-Weak Correlation)

COs	**Upon completion of course the students will be able to**	**PO 4**	**PO 5**	**PO 9**	**PSO 2**
CO1	install RPA packages	3	3	3	3
CO2	apply variables, data types, control statements in designing RPA	3	3	3	3
CO3	make use of data manipulation, recording and scrapping techniques	3	3	3	3
CO4	use selectors, data tables in excel for automation	3	3	3	3
CO5	develop email and PDF automation	3	3	3	3

List of Experiments

Week	**Title/Experiment**
1	Installation of RPA packages.
2	Perform automation for variables and data types.
3	Design a process for control flow: a) Conditional Statements b) Iteration
4	Create a process for data manipulation - scalar variables, collections, tables, text manipulation.
5	Design a process for recording-basic, desktop and web.
6	Design a process for scrapping: a) Screen scrapping b) Data scrapping
7	Perform automation for customizing the Selectors.
8	Create a process for image and text automation.
9	Design a process for automating Data tables in Excel.
10	Perform email automation.
11	Design a process to read all PDF files from a folder and then close them all.
12	Create an automation to change the background color of excel cell/range in Ui Path.
13	Design a process to Generate Covid-19 report and send this report to the required recipient.
14	Create a Process which reminds a user to take his medicine after every 4Hr.

References

1. Robotic Process Automation Lab Manual, Department of CSE, CMRIT, Hyd.

Micro-Projects: Student should submit a report on one of the following/any other micro-project(s) approved by the lab faculty before commencement of lab internal examination.

1. Web Scraping.
2. Data Migration.
3. CRM Upgrading.
4. Call Center Operations.
5. On-boarding Employees.
6. Payroll Processing.
7. Legal Process.
8. Data Wiring for Healthcare.
9. Claims Processing.
10. Support Sales and Marketing Process.

INDEX

INTRODUCTION:

UiPath is a leading robotic process automation (RPA) software platform designed to automate repetitive tasks, streamline workflows, and improve operational efficiency across a wide range of industries. It provides a comprehensive suite of tools and capabilities that enable organizations to create, deploy, manage, and optimize software robots, known as "bots," to automate business processes.

Here's an introduction to some key aspects of UiPath:

- Robotic Process Automation (RPA): RPA is a technology that uses software robots or "bots" to automate repetitive, rule-based tasks traditionally performed by humans. These tasks can include data entry, data extraction, document processing, report generation, and more. UiPath enables organizations to automate these tasks without requiring changes to existing systems or IT infrastructure.
- UiPath Platform: The UiPath platform consists of several components that work together to enable end-to-end automation:
  - UiPath Studio: A visual design environment where users can create automation workflows using a drag-and-drop interface. It provides a wide range of pre-built activities and integrations with other systems and applications.
  - UiPath Robot: The execution engine that runs automation workflows created in UiPath Studio. Robots can be deployed on desktops, virtual machines, or servers, and can interact with various applications and systems to perform automated tasks.
  - UiPath Orchestrator: A centralized platform for managing and monitoring automation processes across the organization. It provides features such as scheduling, monitoring, logging, and reporting, as well as user and robot management capabilities.
  - UiPath Assistant: A lightweight desktop application that provides a simplified interface for end-users to interact with and trigger automation tasks created in UiPath Studio.
- Features and Capabilities: UiPath offers a wide range of features and capabilities to support automation initiatives:
  - Citizen Development: UiPath's low-code/no-code capabilities enable business users with limited technical skills to create and deploy automation workflows.
  - Integration: UiPath integrates with a variety of applications, databases, and systems, allowing users to automate end-to-end processes that span multiple systems.
  - AI and Machine Learning: UiPath leverages artificial intelligence (AI) and machine learning (ML) technologies to enhance automation capabilities, such as natural language processing (NLP) for document understanding and computer vision for automating tasks involving unstructured data.
  - Security and Governance: UiPath provides robust security features and governance controls to ensure the confidentiality, integrity, and availability of automation workflows and data.
- Use Cases: UiPath is used across various industries and functions to automate a wide range of business processes, including finance and accounting, human resources, customer service, supply chain management, and more. Common use cases include invoice processing, data entry, customer onboarding, report generation, and data migration.

Overall, UiPath empowers organizations to accelerate digital transformation, improve operational efficiency, reduce costs, and drive innovation through automation. Its user-friendly interface, extensive capabilities, and scalability make it a popular choice for businesses of all sizes looking to leverage the power of RPA.

WEEK-1

EXPERIMENT NO: 1. UiPath installation steps:

Installing UiPath involves several steps. Here's a detailed guide on how to install UiPath Studio:

1. **Download UiPath Studio**:
    - Go to the UiPath website (https://www.uipath.com/start-trial).
    - Click on "Start Free" or "Download Community Edition" to download UiPath Studio.
2. **Run the Installer**:
    - Once the download is complete, run the installer by double-clicking on the downloaded file (e.g., UiPathStudioSetup.exe).
3. **Select Installation Type**:
    - Choose the installation type:
        - **Enterprise** (for licensed users with an Enterprise plan)
        - **Community** (for free Community Edition)
    - Click "Continue" or "Next" to proceed.
4. **Accept License Agreement**:
    - Read and accept the license agreement.
    - Click "Next" to continue.
5. **Choose Installation Folder**:
    - Select the folder where you want to install UiPath Studio.
    - Click "Next" to proceed.
6. **Select Components**:
    - Choose the components you want to install:
        - UiPath Studio: The main development environment.
        - UiPath Robot: The execution agent for running automation processes.
        - UiPath Orchestrator: The web-based management platform (optional).
    - Click "Next" to continue.
7. **Select Additional Tasks** (Optional):
    - Choose any additional tasks you want to perform, such as creating shortcuts or installing dependencies.
    - Click "Install" to start the installation process.
8. **Installation Progress**:
    - Wait for the installation to complete. This may take a few minutes.
9. **Complete Installation**:
    - Once the installation is finished, click "Finish" to exit the installer.
10. **Activate License** (for Enterprise Users):
    - If you're using the Enterprise edition, you'll need to activate your license:
        - Open UiPath Studio.
        - Click on "Help" in the top menu.
        - Select "Activate License."
        - Enter your license key and follow the prompts to activate.
11. **Launch UiPath Studio**:
    - Double-click on the UiPath Studio icon on your desktop or search for it in the Start menu.
    - If you're using the Community Edition, you'll need to sign in with your UiPath account.
12. **Update Packages** (Optional):
    - Once UiPath Studio is launched, you can update packages by clicking on the "Manage Packages" button in the "Design" tab.
    - Update or install any required packages for your projects.

That's it! You've successfully installed UiPath Studio. You can now start creating, testing, and executing automation projects using UiPath

WEEK-2

EXPERIMENT NO: 2. **Perform automation for variables and data types:**

In UiPath, you can perform automation tasks involving variables and data types using various activities within workflows. Here's how you can work with variables and data types in UiPath:

1. **Create Variables**:
   - In UiPath Studio, you can create variables by going to the "Variables" panel in the bottom left corner.
   - Click on the "Create Variable" button and define the variable name, data type, and default value if needed.
   - UiPath supports various data types such as String, Integer, Double, Boolean, Array, DataTable, etc.
2. **Assign Values to Variables**:
   - Use the "Assign" activity to assign values to variables.
   - Drag and drop the "Assign" activity onto the workflow canvas.
   - In the "To" field, select the variable to which you want to assign a value.
   - In the "Value" field, specify the value to be assigned, which can be a constant value or another variable.
3. **Perform Operations with Variables**:
   - Use activities such as "Assign," "Invoke Method," or "Invoke Code" to perform operations with variables.
   - You can perform arithmetic operations, string manipulations, comparisons, etc., using variables.
4. **Use Data Types**:
   - Depending on your automation requirements, you may need to work with different data types.
   - UiPath provides activities to convert between different data types. For example, use the "Convert.ToDouble" activity to convert a String to Double.
   - Use activities like "Build DataTable" to create and work with DataTables.
5. **Iterate Through Data**:
   - Use loops such as "For Each" or "While" to iterate through collections or arrays.
   - Use variables as loop counters or to store iterated values.
6. **Handle Data from External Sources**:
   - Use activities such as "Excel Application Scope" or "Read Range" to read data from Excel files.
   - Use activities such as "Get Outlook Mail Messages" to retrieve emails from Outlook.
   - Store data in variables or data tables for further processing.
7. **Debugging and Validation**:
   - Use the "Write Line" activity to output variable values during debugging.
   - Use breakpoints to pause execution and inspect variable values.
8. **Error Handling**:
   - Use Try-Catch blocks to handle exceptions when working with variables and data types.
   - Use activities such as "Throw" or "Log Message" to log errors or exceptions.
9. **Testing**:
   - Test your automation workflows with different data sets to ensure they handle various scenarios correctly.

By following these steps, you can effectively work with variables and data types in UiPath to create robust and efficient automation workflows

WEEK-3

EXPERIMENT NO: 3. **Design a process for control flow: Conditional Statements:**

Designing a process with conditional statements involves creating a workflow that executes different actions based on specific conditions. Let's create a simple process using UiPath that includes conditional statements. In this example, we'll design a process that checks whether a given number is positive, negative, or zero.

Step 1: Open UiPath Studio

- Launch UiPath Studio and create a new project or open an existing one.

Step 2: Add Variables

- Create a variable to store the number. Go to the "Variables" panel, click on "Create Variable," and name it inputNumber with the data type Int32.

Step 3: Input Dialog

- Drag and drop an "Input Dialog" activity onto the workflow canvas.
- Configure the activity to prompt the user to enter a number and store the result in the inputNumber variable.

Step 4: Assign Activity (Convert to Int32)

- Drag and drop an "Assign" activity below the "Input Dialog."
- Assign the value of inputNumber to itself converted to Int32. Use the following expression:

makefileCopy code

```
inputNumber = Convert.ToInt32(inputNumber)
```

Step 5: Flow Decision Activity

- Drag and drop a "Flow Decision" activity onto the canvas.
- Configure the condition to check whether the inputNumber is greater than 0.
- For the "True" branch, add activities to display a message indicating the number is positive.
- For the "False" branch, add another "Flow Decision" to check if the number is equal to 0.
  - For the "True" branch, display a message indicating the number is zero.
  - For the "False" branch, display a message indicating the number is negative.

Step 6: Display Messages

- Use the "Message Box" activity to display messages for positive, zero, and negative numbers.
  - For positive numbers: "The number is positive."
  - For zero: "The number is zero."
  - For negative numbers: "The number is negative."

Step 7: Finalize the Workflow

- Connect the activities in a logical sequence to form the complete workflow.
- Save your workflow and run it to test the conditional statements.

Example Workflow (Simplified representation):

plaintextCopy code

```
[Start] ---> [Input Dialog] ---> [Assign (Convert to Int32)] ---> [Flow Decision (Is Positive?)] | | V | [Message Box (Positive)] | | [Flow Decision (Is Zero?)] | | V | [Message Box (Zero)] | | [Message Box (Negative)] [End]
```

Notes:

- You can enhance this process based on your specific requirements.
- Use additional activities and logic for error handling or more complex scenarios.
- This example focuses on the fundamental structure of a conditional statement workflow in UiPath.

WEEK-4

EXPERIMENT NO: 4 . **Create a process for data manipulation-scalar variables, collections, tables, text-manipulation**:

Creating a process for data manipulation involving scalar variables, collections, tables, and text manipulation involves several steps. Below is a general outline of how you can design such a process:

1. **Identify Data Sources**:
    - Determine the source(s) of data you'll be working with. This could be Excel files, databases, web pages, or any other source.
2. **Input Data Retrieval**:
    - Use UiPath activities to retrieve data from the identified source(s). For example, if the data is in an Excel file, use activities like "Excel Application Scope" and "Read Range" to read the data into a DataTable variable.
3. **Scalar Variables Manipulation**:
    - Perform any necessary operations on scalar variables (individual data points) using UiPath activities such as Assign, Invoke Method, or Write Line.
4. **Collections Manipulation**:
    - Manipulate collections (lists, arrays, dictionaries) as needed. Use activities such as Add to Collection, Remove from Collection, For Each, or LINQ queries to manipulate collections of data.
5. **Tables Manipulation**:
    - Manipulate DataTables or other table-like structures using activities such as Filter DataTable, Join DataTable, or Write Range.
6. **Text Manipulation**:
    - Perform text manipulation tasks using activities such as String Manipulation, Regular Expressions, or Split.
7. **Output Data Handling**:
    - Determine how you want to output the manipulated data. This could involve writing data back to the same or different sources, displaying it in a user interface, or exporting it to a report.
8. **Error Handling**:
    - Implement error handling mechanisms using Try Catch activities to handle exceptions gracefully and ensure the robustness of your process.
9. **Testing and Debugging**:
    - Test your process thoroughly with different sets of input data to ensure it behaves as expected. Debug any issues that arise during testing.

10. **Documentation and Maintenance**:

    - Document your process thoroughly, including any assumptions made and steps taken. This documentation will be helpful for future reference or when handing over the process to someone else. Also, periodically review and update the process to accommodate any changes in requirements or technology.

11. **Optimization**:

    - Optimize your process for performance and efficiency by identifying and eliminating any redundant or unnecessary steps. Use UiPath best practices and consider factors like memory usage and execution time.

12. **Deployment**:

    - Deploy your process to the desired environment (e.g., UiPath Studio, Orchestrator) for execution.

By following these steps, you can create a robust and efficient process for data manipulation using UiPath, covering scalar variables, collections, tables, and text manipulation

WEEK-5

EXPERIMENT NO: 5. **Design a process for recording-basic, desktop and web:**

Designing a process for recording interactions in basic, desktop, and web applications using UiPath involves the following steps:

1. **Identify the Application**: Determine the applications (basic, desktop, and web) that you want to record interactions with.
2. **Set Up Your Environment**:
   - Install UiPath Studio on your machine if you haven't already.
   - Ensure that the necessary browsers (for web recording) and desktop applications are installed and accessible.
3. **Create a New Project**:
   - Open UiPath Studio and create a new project for your recording task.
4. **Recording Basic Interactions**:
   - For basic interactions, such as keyboard inputs or mouse clicks:
     - Use the "Basic Recording" feature in UiPath Studio.
     - Open the application you want to interact with.
     - Start the recording and perform the desired actions within the application.
     - Stop the recording when finished.
5. **Recording Desktop Interactions**:
   - For desktop applications:
     - Use the "Desktop Recording" feature in UiPath Studio.
     - Open the desktop application you want to interact with.
     - Start the recording and perform the desired actions within the application.
     - Stop the recording when finished.
6. **Recording Web Interactions**:
   - For web applications:
     - Use the "Web Recording" feature in UiPath Studio.
     - Open the web browser and navigate to the web application you want to interact with.
     - Start the recording and perform the desired actions within the web application.
     - Stop the recording when finished.

7. **Refinement and Validation**:
    - Review the recorded interactions to ensure accuracy and completeness.
    - Edit and refine the recorded steps as needed using UiPath Studio's workflow editor.
    - Validate the workflow by running it and observing the interactions with the applications.
8. **Error Handling**:
    - Implement error handling mechanisms within the workflow to handle exceptions gracefully, such as using Try Catch activities.
9. **Testing and Debugging**:
    - Test the recorded workflow with different scenarios to ensure it behaves as expected.
    - Debug any issues that arise during testing and refine the workflow accordingly.
10. **Documentation and Maintenance**:
    - Document the recorded workflow, including any assumptions made and steps taken.
    - Maintain the workflow by updating it as needed to accommodate changes in the applications or business requirements.
11. **Deployment**:
    - Deploy the recorded workflow to the desired environment (e.g., UiPath Studio, Orchestrator) for execution.

By following these steps, you can design a process for recording interactions in basic, desktop, and web applications using UiPath.

WEEK-6

EXPERIMENT NO: 6.**Designing a process for scraping data using screen scraping and data scraping**

Designing a process for scraping data using screen scraping and data scraping in UiPath involves several steps:

1. **Identify Data Source**:
    - Determine the application or website from which you want to scrape data.
2. **Screen Scraping**:
    - Screen scraping is used when the data you want to extract is not available in a structured format and requires extracting text or images from the screen.
    - Use the "Screen Scraping" wizard in UiPath Studio:
        - Open UiPath Studio and create a new project.
        - Use the "Screen Scraping" wizard to indicate the elements on the screen from which you want to extract data (e.g., text, images).
        - Configure the scraping options (e.g., OCR engine, scraping method) based on the type of data and its location on the screen.
        - Extract the data and store it in variables or write it to a file or database.
3. **Data Scraping**:
    - Data scraping is used when the data you want to extract is structured and can be extracted from tables or lists on a web page or application.
    - Use the "Data Scraping" wizard in UiPath Studio:
        - Open UiPath Studio and create a new project.
        - Use the "Data Scraping" wizard to indicate the table or list from which you want to extract data.
        - Configure the scraping options (e.g., column headers, next page navigation) to ensure all desired data is captured.
        - Extract the data and store it in a DataTable variable or write it to a file or database.
4. **Data Manipulation (Optional)**:
    - Perform any necessary data manipulation or cleaning operations on the extracted data using UiPath activities or custom scripts:
        - For example, you can use activities like Filter DataTable, Join DataTable, or Write Range to manipulate DataTables.

    - For text manipulation, you can use activities like String Manipulation or Regular Expressions.

5. **Error Handling**:
    - Implement error handling mechanisms using Try Catch activities to handle exceptions gracefully, especially when dealing with unreliable or inconsistent data sources.
6. **Testing and Debugging**:
    - Test the scraping process with different scenarios to ensure it extracts the data accurately and reliably.
    - Debug any issues that arise during testing and refine the process accordingly.
7. **Documentation and Maintenance**:
    - Document the scraping process, including any assumptions made and steps taken.
    - Maintain the process by updating it as needed to accommodate changes in the data source or structure.
8. **Deployment**:
    - Deploy the scraping process to the desired environment (e.g., UiPath Studio, Orchestrator) for execution.

By following these steps, you can design a process for scraping data using screen scraping and data scraping in UiPath, enabling you to extract structured and unstructured data from various sources

WEEK-7

EXPERIMENT NO: 7 . P**erform automation for customizing selectors.**

To perform automation for customizing selectors, you can follow these steps using UiExplorer and editing selectors in the properties of activities:

1. **Identify the Target Element**:
   - Open UiPath Studio and create a new project or open an existing one.
   - Identify the UI element you want to interact with in your automation process.
2. **Use UiExplorer to Customize Selector**:
   - Click on the "Indicate element inside browser" or "Indicate element inside window" option in the UiPath Studio toolbar.
   - Hover your mouse over the target element, and UiExplorer will highlight it.
   - Click on the target element to inspect its properties in UiExplorer.
3. **Customize Selector Properties**:
   - In UiExplorer, review the properties of the selected element.
   - Customize the selector properties as needed to ensure a robust and reliable selection.
   - Adjust the selector to include unique attributes that identify the element, avoiding properties that may change dynamically.
   - Use wildcards (*) where necessary to handle dynamic attribute values.
   - Remove unnecessary or volatile properties from the selector to improve stability.
4. **Validate Selector**:
   - After customizing the selector, click on the "Validate" button in UiExplorer to ensure that the selector correctly identifies the target element.
   - Verify that the selector is unique and does not match unintended elements on the screen.
5. **Update Selector in Activity Properties**:
   - Close UiExplorer and return to UiPath Studio.
   - Open the activity (e.g., Click, Type Into) that interacts with the target element.
   - In the properties pane of the activity, locate the "Selector" field.
   - Paste the customized selector into the "Selector" field to replace the default selector.
6. **Test the Automation**:
   - Test the automation process to verify that the customized selector correctly identifies and interacts with the target element.
   - Run the workflow and observe the behavior of the activities interacting with the element.

7. **Error Handling** (Optional):
    - Implement error handling mechanisms to handle scenarios where the target element cannot be found or interactions fail.
    - Use Try Catch activities or Element Exists activities to handle exceptions gracefully and provide appropriate error messages.
8. **Documentation and Maintenance**:
    - Document the customized selector and any assumptions made during the customization process.
    - Maintain the automation project by periodically reviewing and updating selectors as needed, especially when the application undergoes changes.

By following these steps, you can effectively customize selectors in UiPath to create robust and reliable automation processes that accurately interact with target elements in desktop or web applications

WEEK-8

EXPERIMENT NO: 8. **Create a process for image and text automation:**

To create a process for image and text automation in UiPath, we'll design a simple workflow that involves extracting text from an image using OCR (Optical Character Recognition) and performing text manipulation tasks. Below are the steps to create this process:

1. **Open UiPath Studio**:
   - Launch UiPath Studio on your computer.
2. **Create a New Project**:
   - Click on "New Process" to create a new automation project.
   - Give your project a name and choose a location to save it.
3. **Design the Workflow**:
   - Drag and drop a "Sequence" onto the canvas. This will be the main container for your activities.
4. **Add Activities for Image and Text Automation**:
   - Add activities to perform image and text automation tasks. Below are the activities you'll use:
   - **Load Image**: Use the "Load Image" activity to load the image file from which you want to extract text.
   - **OCR Text Recognition**: Use the "OCR Text Recognition" activity to extract text from the loaded image using OCR technology. Configure the OCR engine and other settings as per your requirement.
   - **Assign Activity**: Use the "Assign" activity to manipulate the extracted text. For example, you can assign the text to a variable and then manipulate it using string manipulation functions or regular expressions.
   - **Write Line**: Use the "Write Line" activity to output the manipulated text to the Output panel or a text file for verification.
5. **Configure Activities**:
   - Configure each activity with the necessary inputs and properties. For example:
     - Specify the file path for the image in the "Load Image" activity.
     - Choose the appropriate OCR engine and language in the "OCR Text Recognition" activity.
     - Write the necessary manipulation code in the "Assign" activity to process the extracted text.
6. **Run the Workflow**:
   - Save your workflow and click on the "Run" button in UiPath Studio to execute the automation process.
   - Monitor the execution and verify that the text is extracted from the image and manipulated as expected.
7. **Error Handling** (Optional):
   - Implement error handling mechanisms using Try Catch activities to handle exceptions gracefully, especially during image loading or text extraction.
8. **Documentation and Maintenance**:
   - Document your automation process, including any assumptions made and steps taken.
   - Maintain the process by updating it as needed to accommodate changes in the image or text extraction requirements.
9. **Deployment**:
   - Deploy the automation process to the desired environment (e.g., UiPath Studio, Orchestrator) for execution.

By following these steps, you can create a process for image and text automation in UiPath, enabling you to extract text from images and perform text manipulation tasks as part of your automation workflows

WEEK-9

EXPERIMENT NO: 9. **Design a process for automating Data tables in Excel**

To design a process for automating Data tables in Excel using UiPath, you can follow these steps:

1. **Open UiPath Studio**:
    - Launch UiPath Studio on your computer.
2. **Create a New Project**:
    - Click on "New Process" to create a new automation project.
    - Give your project a name and choose a location to save it.
3. **Design the Workflow**:
    - Drag and drop a "Sequence" onto the canvas. This will be the main container for your activities.
4. **Add Activities for Excel Automation**:
    - Add activities to perform automation tasks related to Data tables in Excel. Below are the activities you'll use:
    - **Excel Application Scope**: Use the "Excel Application Scope" activity to specify the Excel file you want to work with. This activity will open Excel and establish a connection to the file.
    - **Read Range**: Use the "Read Range" activity to read data from a specific range of cells in the Excel file and store it in a Data table variable.
    - **Write Range**: Use the "Write Range" activity to write data from a Data table variable back to a specific range of cells in the Excel file.
    - **Filter Data table**: Use the "Filter Data table" activity to filter data in the Data table variable based on specified criteria.
    - **Join Data table**: Use the "Join Data table" activity to join two Data table variables based on a common key column.
    - **Excel Close Workbook**: Use the "Excel Close Workbook" activity to close the Excel file once the automation process is completed.
5. **Configure Activities**:
    - Configure each activity with the necessary inputs and properties. For example:
        - Specify the file path for the Excel file in the "Excel Application Scope" activity.
        - Specify the range of cells to read from or write to in the "Read Range" and "Write Range" activities.
        - Specify the filter criteria in the "Filter DataTable" activity.
        - Specify the common key column for joining DataTables in the "Join DataTable" activity.
6. **Run the Workflow**:
    - Save your workflow and click on the "Run" button in UiPath Studio to execute the automation process.
    - Monitor the execution and verify that the DataTables in Excel are automated as expected.
7. **Error Handling** (Optional):
    - Implement error handling mechanisms using Try Catch activities to handle exceptions gracefully, especially during Excel file operations.
8. **Documentation and Maintenance**:
    - Document your automation process, including any assumptions made and steps taken.
    - Maintain the process by updating it as needed to accommodate changes in the Excel file structure or automation requirements.
9. **Deployment**:
    - Deploy the automation process to the desired environment (e.g., UiPath Studio, Orchestrator) for execution.

By following these steps, you can design a process for automating DataTables in Excel using UiPath, enabling you to read, write, filter, and join data in Excel files as part of your automation workflows

WEEK-10

EXPERIMENT NO: 10. **Performing email automation:**

Performing email automation typically involves tasks such as sending emails, reading emails, and processing email attachments. Below, I'll outline a simple email automation process using UiPath:

1. **Open UiPath Studio**:
   - Launch UiPath Studio on your computer.
2. **Create a New Project**:
   - Click on "New Process" to create a new automation project.
   - Give your project a name and choose a location to save it.
3. **Design the Workflow**:
   - Drag and drop a "Sequence" onto the canvas. This will be the main container for your activities.
4. **Add Activities for Email Automation**:
   - Add activities to perform email automation tasks. Below are the activities you'll use:
   - **Get IMAP Mail Messages**: Use the "Get IMAP Mail Messages" activity to retrieve emails from an IMAP mailbox. Configure the activity to connect to your email server and specify the mailbox and search criteria (e.g., unread emails).
   - **For Each**: Use a "For Each" activity to iterate through each retrieved email message.
   - **Send SMTP Mail Message**: Use the "Send SMTP Mail Message" activity to send an email. Configure the activity with the necessary details such as sender, recipient, subject, and body.
   - **Save Attachment**: Use the "Save Attachment" activity to save attachments from email messages to a specified location on your computer.
5. **Configure Activities**:
   - Configure each activity with the necessary inputs and properties. For example:
     - Specify the IMAP server settings and mailbox credentials in the "Get IMAP Mail Messages" activity.
     - Specify the email details (sender, recipient, subject, body) in the "Send SMTP Mail Message" activity.
     - Specify the folder path to save attachments in the "Save Attachment" activity.
6. **Run the Workflow**:
   - Save your workflow and click on the "Run" button in UiPath Studio to execute the email automation process.
   - Monitor the execution and verify that emails are retrieved, sent, and attachments are saved as expected.
7. **Error Handling** (Optional):
   - Implement error handling mechanisms using Try Catch activities to handle exceptions gracefully, especially during email operations.
8. **Documentation and Maintenance**:
   - Document your email automation process, including any assumptions made and steps taken.
   - Maintain the process by updating it as needed to accommodate changes in email server settings or automation requirements.
9. **Deployment**:
   - Deploy the email automation process to the desired environment (e.g., UiPath Studio, Orchestrator) for execution.

By following these steps, you can perform email automation using UiPath, enabling you to send emails, retrieve emails, and process email attachments as part of your automation workflows.

WEEK-11

EXPERIMENT NO: 11. **Design a process to read all PDF files from a folder and then close them all:**

To design a process to read all PDF files from a folder and then close them all using UiPath, you can follow these steps:

1. **Open UiPath Studio**:
    - Launch UiPath Studio on your computer.
2. **Create a New Project**:
    - Click on "New Process" to create a new automation project.
    - Give your project a name and choose a location to save it.
3. **Design the Workflow**:
    - Drag and drop a "Sequence" onto the canvas. This will be the main container for your activities.
4. **Add Activities for PDF Automation**:
    - Add activities to perform PDF automation tasks. Below are the activities you'll use:
    - **Assign Activity**: Use the "Assign" activity to specify the folder path where your PDF files are located. Store this path in a variable.
    - **Directory.GetFiles**: Use the "Directory.GetFiles" method to get a list of all PDF files in the specified folder. Store this list in an array variable.
    - **For Each**: Use a "For Each" activity to iterate through each PDF file in the array.
    - **Read PDF Text**: Use the "Read PDF Text" activity to read the text from each PDF file.
    - **Close Application**: Use the "Close Application" activity to close the PDF reader application (e.g., Adobe Acrobat) after reading each PDF file.
5. **Configure Activities**:
    - Configure each activity with the necessary inputs and properties. For example:
        - Specify the folder path variable in the "Assign" activity to define the location of your PDF files.
        - Specify the array variable containing the list of PDF files in the "For Each" activity.
        - Configure the "Read PDF Text" activity to read the text from each PDF file.
6. **Run the Workflow**:
    - Save your workflow and click on the "Run" button in UiPath Studio to execute the PDF automation process.
    - Monitor the execution and verify that the text is read from each PDF file, and the PDF reader application is closed after processing each file.
7. **Error Handling** (Optional):
    - Implement error handling mechanisms using Try Catch activities to handle exceptions gracefully, especially during PDF file operations.
8. **Documentation and Maintenance**:
    - Document your PDF automation process, including any assumptions made and steps taken.
    - Maintain the process by updating it as needed to accommodate changes in the folder structure or PDF file format.

9. **Deployment**:
    - Deploy the PDF automation process to the desired environment (e.g., UiPath Studio, Orchestrator) for execution.

By following these steps, you can design a process to read all PDF files from a folder and then close them all using UiPath, enabling you to extract text from multiple PDF files as part of your automation workflows.

WEEK-12

EXPERIMENT NO: 12. **Create an automation to change the background Color of excel cell/range in UiPath:**

To create an automation to change the background color of an Excel cell or range using UiPath, you can follow these steps:

1. **Open UiPath Studio**:
   - Launch UiPath Studio on your computer.
2. **Create a New Project**:
   - Click on "New Process" to create a new automation project.
   - Give your project a name and choose a location to save it.
3. **Design the Workflow**:
   - Drag and drop a "Sequence" onto the canvas. This will be the main container for your activities.
4. **Add Activities for Excel Automation**:
   - Add activities to perform Excel automation tasks. Below are the activities you'll use:
   - **Excel Application Scope**: Use the "Excel Application Scope" activity to specify the Excel file you want to work with. This activity will open Excel and establish a connection to the file.
   - **Write Cell**: Use the "Write Cell" activity to write data to a specific cell in the Excel file. This activity can also be used to change the background color of a cell.
5. **Configure Activities**:
   - Configure each activity with the necessary inputs and properties. For example:
     - Specify the file path for the Excel file in the "Excel Application Scope" activity.
     - Specify the cell address (e.g., "A1") in the "Write Cell" activity where you want to change the background color.
     - Use the **Background** property of the "Write Cell" activity to set the desired background color. You can use color codes (e.g., "#FF0000" for red) or named colors (e.g., "Red").
6. **Run the Workflow**:
   - Save your workflow and click on the "Run" button in UiPath Studio to execute the Excel automation process.
   - Monitor the execution and verify that the background color of the specified cell is changed as expected.
7. **Error Handling** (Optional):
   - Implement error handling mechanisms using Try Catch activities to handle exceptions gracefully, especially during Excel file operations.
8. **Documentation and Maintenance**:
   - Document your Excel automation process, including any assumptions made and steps taken.
   - Maintain the process by updating it as needed to accommodate changes in the Excel file structure or automation requirements.
9. **Deployment**:
   - Deploy the Excel automation process to the desired environment (e.g., UiPath Studio, Orchestrator) for execution.

By following these steps, you can create an automation to change the background color of an Excel cell or range using UiPath, enabling you to format Excel files as part of your automation workflows.

WEEK-13

EXPERIMENT NO: 13. **Design a process to Generate Covid-19 report and send this report to the required recipient:**

To design a process to generate a COVID-19 report and send it to the required recipient using UiPath, you can follow these steps:

1. **Open UiPath Studio**:
    - Launch UiPath Studio on your computer.
2. **Create a New Project**:
    - Click on "New Process" to create a new automation project.
    - Give your project a name and choose a location to save it.
3. **Design the Workflow**:
    - Drag and drop a "Sequence" onto the canvas. This will be the main container for your activities.
4. **Add Activities for Report Generation**:
    - Add activities to generate the COVID-19 report. Below are the activities you'll use:
    - **Data Scraping**: Use the "Data Scraping" wizard or custom activities to scrape COVID-19 data from reliable sources (e.g., government websites, APIs). Extract relevant data such as number of cases, deaths, recoveries, etc.
    - **Create Report**: Use activities like "Build Data Table" to structure the extracted data into a report format (e.g., Excel, HTML).
5. **Add Activities for Email Automation**:
    - Add activities to send the report to the required recipient. Below are the activities you'll use:
    - **Send SMTP Mail Message**: Use the "Send SMTP Mail Message" activity to send an email. Configure the activity with the necessary details such as sender, recipient, subject, body, and attachment (attach the generated report).
6. **Configure Activities**:
    - Configure each activity with the necessary inputs and properties. For example:
        - Specify the data source and extraction method in the "Data Scraping" activity.
        - Structure the extracted data into a report format using the "Build Data Table" activity.
        - Configure the "Send SMTP Mail Message" activity with the appropriate email server settings and recipient details.
7. **Run the Workflow**:
    - Save your workflow and click on the "Run" button in UiPath Studio to execute the automation process.
    - Monitor the execution and verify that the COVID-19 report is generated and sent to the required recipient.
8. **Error Handling** (Optional):
    - Implement error handling mechanisms using Try Catch activities to handle exceptions gracefully, especially during data scraping or email sending operations.
9. **Documentation and Maintenance**:
    - Document your automation process, including any assumptions made and steps taken.
    - Maintain the process by updating it as needed to accommodate changes in data sources or automation requirements.
10. **Deployment**:
    - Deploy the automation process to the desired environment (e.g., UiPath Studio, Orchestrator) for execution.

By following these steps, you can design a process to generate a COVID-19 report and send it to the required recipient using UiPath, enabling you to automate the reporting process and ensure timely dissemination of information.

WEEK-14

EXPERIMENT NO: 14. **Create a process which reminds a user to take his medicine after every 4 hours:**

To create a process that reminds a user to take their medicine every 4 hours using UiPath, you can follow these steps:

1. **Open UiPath Studio**:
   - Launch UiPath Studio on your computer.
2. **Create a New Project**:
   - Click on "New Process" to create a new automation project.
   - Give your project a name and choose a location to save it.
3. **Design the Workflow**:
   - Drag and drop a "Sequence" onto the canvas. This will be the main container for your activities.
4. **Add Activities for Reminder**:
   - Add activities to remind the user to take their medicine. Below are the activities you'll use:
   - **Delay**: Use the "Delay" activity to wait for 4 hours before reminding the user. Set the duration to 4 hours (e.g., 14400 seconds).
   - **Message Box**: Use the "Message Box" activity to display a reminder message to the user. Customize the message to indicate that it's time to take their medicine.
5. **Run the Workflow**:
   - Save your workflow and click on the "Run" button in UiPath Studio to execute the automation process.
   - The automation will wait for 4 hours and then display a message box reminding the user to take their medicine.
6. **Repeat the Process**:
   - To ensure continuous reminders, you can add a loop around the sequence of activities so that the process repeats indefinitely.
   - Use the "While" or "Do While" activity to create a loop that continuously executes the reminder activities every 4 hours.
7. **Error Handling** (Optional):
   - Implement error handling mechanisms using Try Catch activities to handle exceptions gracefully, especially during delay operations.
8. **Documentation and Maintenance**:
   - Document your automation process, including any assumptions made and steps taken.
   - Maintain the process by updating it as needed to accommodate changes in reminder messages or automation requirements.
9. **Deployment**:
   - Deploy the automation process to the desired environment (e.g., UiPath Studio, Orchestrator) for execution.

By following these steps, you can create a process that reminds a user to take their medicine every 4 hours using UiPath, helping to ensure medication adherence and better health outcomes.

References:

- ➢ https://docs.uipath.com/studio/standalone/2022.10/user-guide/install-studio