PDF Splitter BOT

Submitted by,

VIGNESHWARI E
CSE, 3rd Year

PDF Splitter BOT in UiPath Studio

Problem Statement:

1. Read data from PDF and identify the document
2. The input PDF document has 3 type of documents: Invoice, Receipts, and Clinic bills.
3. Identify the document type split the page from PDF and save it in the individual folder.

Introduction:

In the modern business world, handling large volumes of documents such as invoices, receipts, and clinic bills is a common challenge. These documents often come in a single PDF file containing multiple pages, making manual sorting and organizing a tedious and time-consuming task. Moreover, errors in manual handling can lead to misplacement of important records, affecting the overall efficiency and productivity of the organization.

To address this challenge, the **PDF Splitter BOT** project was developed using UiPath, a powerful robotic process automation (RPA) platform. This bot automates the process of reading, identifying, splitting, and saving PDF documents into categorized folders based on their type. By leveraging UiPath's PDF activities and OCR capabilities, the bot ensures accurate classification of documents and significantly reduces human intervention.

The **PDF Splitter BOT** is designed to handle three types of documents: **Invoices**, **Receipts**, and **Clinic Bills**. It reads the content of each page, identifies the document type using specific keywords, splits the PDF into individual pages, and saves them in their respective folders. This automation not only enhances efficiency but also improves accuracy, making it an invaluable tool for businesses dealing with repetitive document management tasks.

Through this project, we aim to demonstrate the power of RPA in automating repetitive workflows and highlight the practical applications of UiPath in streamlining document management processes.

Objective:

The primary goal of the PDF Splitter BOT is to automate the process of handling PDF documents, specifically focusing on the categorization, splitting, and saving of PDF files based on their content. In the real world, businesses often deal with a variety of documents, such as invoices, receipts, and clinic bills, which are stored in a single file. This project aims to solve that problem by enabling the automation of sorting and storing these documents into separate folders based on their type. The process involves identifying the document type (Invoice, Receipt, or Clinic Bill) from a given PDF, splitting the pages, and saving each page into the appropriate folder. By automating this task, businesses can save time, reduce errors, and improve organizational efficiency.

Technologies Used:

The PDF Splitter BOT was built using UiPath Studio, which is an automation tool designed for creating and managing workflows that interact with desktop, web, and cloud applications. UiPath offers a drag-and-drop interface that makes it easy to automate processes without writing complex code. The core functionality of the bot relies on UiPath's PDF Activities package, which is specifically designed to handle PDF files. This package allows reading text from PDFs, extracting specific pages, splitting documents, and moving files between directories. OCR (Optical Character Recognition) is used in the case of scanned or image-based PDFs, allowing the bot to extract text from images. The combination of these technologies enables the bot to process a wide variety of PDFs and automate the sorting process seamlessly.

Tools and Libraries:

- **UiPath Studio**: This is the main automation tool used to design, develop, and test the bot. It provides a visual, user-friendly interface where workflows are created using predefined activities. The drag-and-drop functionality makes it easy for users to automate repetitive tasks, and its integration with other tools makes it highly effective for real-world applications.
- **UiPath.PDF.Activities**: This is a library that contains all the necessary activities to work with PDF documents in UiPath. The activities allow users to read text, extract pages, merge files, split PDFs, and much more. This package is essential for handling the PDF files in this project.
- **Tesseract OCR (Optional)**: For documents that are scanned as images, the **Read PDF with OCR** activity is used to extract the text using **Tesseract OCR**, which is an open-source optical character recognition engine. This ensures that even image-based PDFs are processed effectively.

Project Overview:

The **PDF Splitter BOT** is designed to automate the process of organizing and storing PDFs in an efficient manner. The input to the bot is a single PDF document that contains multiple pages, some of which may be invoices, receipts, or clinic bills. The bot performs the following tasks:

1. **Reads the PDF**: The bot reads the content of the input PDF using **Read PDF Text** or **Read PDF with OCR** activities. This step extracts the text from the document, which is then analyzed to determine the document type.
2. **Identifies the Document Type**: Based on the extracted text, the bot uses **If** statements to check for keywords such as "Invoice", "Receipt", or "Clinic Bill" to categorize the document. If the document is identified as an Invoice, it is categorized as such, and similarly for Receipts and Clinic Bills.
3. **Splits the PDF Pages**: The bot uses the **Extract PDF Page Range** activity to split the document into individual pages. This ensures that each page is processed and saved separately.
4. **Saves the Pages**: Finally, each page is saved into one of three folders—**Invoices**, **Receipts**, or **Clinic Bills**—based on the identified document type. This step is done using the **Move File** activity, which ensures that each page is stored in the appropriate folder.

Project Workflow:

The workflow of the **PDF Splitter BOT** is divided into several key steps, each designed to handle specific tasks efficiently:

1. **Install UiPath Studio**: First, UiPath Studio is installed and set up on the system. A new project is created in UiPath Studio, and all necessary dependencies, including **UiPath.PDF.Activities**, are added.
2. **Input PDF File**: The **Read PDF Text** activity is used to extract the text from the input PDF file. If the PDF is image-based, the **Read PDF with OCR** activity is used instead. This allows the bot to handle both text-based and image-based PDFs.
3. **Identify Document Type**: The text extracted from the PDF is analyzed using simple string functions (like pdfText.Contains("Invoice")). The bot categorizes the document into one of three types based on these checks.
4. **Create Folders**: Folders are created for each document type (Invoice, Receipt, and Clinic Bill) using the **Create Folder** activity. These folders are located in a predefined directory on the system.
5. **Split PDF Pages**: The **Extract PDF Page Range** activity is used to split the PDF into individual pages. Each page is saved separately to facilitate easy categorization.
6. **Move Files to Folders**: After splitting, the bot moves each page to the corresponding folder. For example, all pages identified as part of an invoice will be moved to the **Invoices** folder, and so on for receipts and clinic bills.

7. **Test the Workflow**: The workflow is then tested using **Debug** mode to ensure that each step works as expected, with PDFs being split and saved into the correct folders.

Challenges Faced:

1. **OCR Accuracy**: One of the challenges encountered was the inability of OCR to correctly extract text from low-quality or heavily formatted PDFs. In such cases, manual adjustments were needed to ensure proper recognition.
2. **Document Type Identification**: Identifying the correct document type was a challenge when the content or format of the documents was inconsistent. The bot had to be adjusted to check for multiple possible keywords and handle cases where the document was ambiguous.
3. **Handling Large PDFs**: Splitting large PDFs into multiple pages posed a performance challenge. The bot was optimized to handle large files efficiently without overwhelming the system.

Future Improvements:

1. **Advanced Document Type Recognition**: Implement machine learning models for more advanced document classification, improving accuracy when identifying different document types.
2. **Error Handling**: Add error-handling mechanisms to ensure the bot can handle cases where the PDF is not readable or is in an unsupported format.
3. **Multi-Language Support**: Enhance OCR capabilities to support multiple languages, ensuring that the bot can handle documents in different languages (e.g., English, Spanish, etc.).
4. **Web Integration**: Integrate the bot with cloud storage services (e.g., Google Drive, Dropbox) to automatically upload the sorted PDFs.

Screenshot:

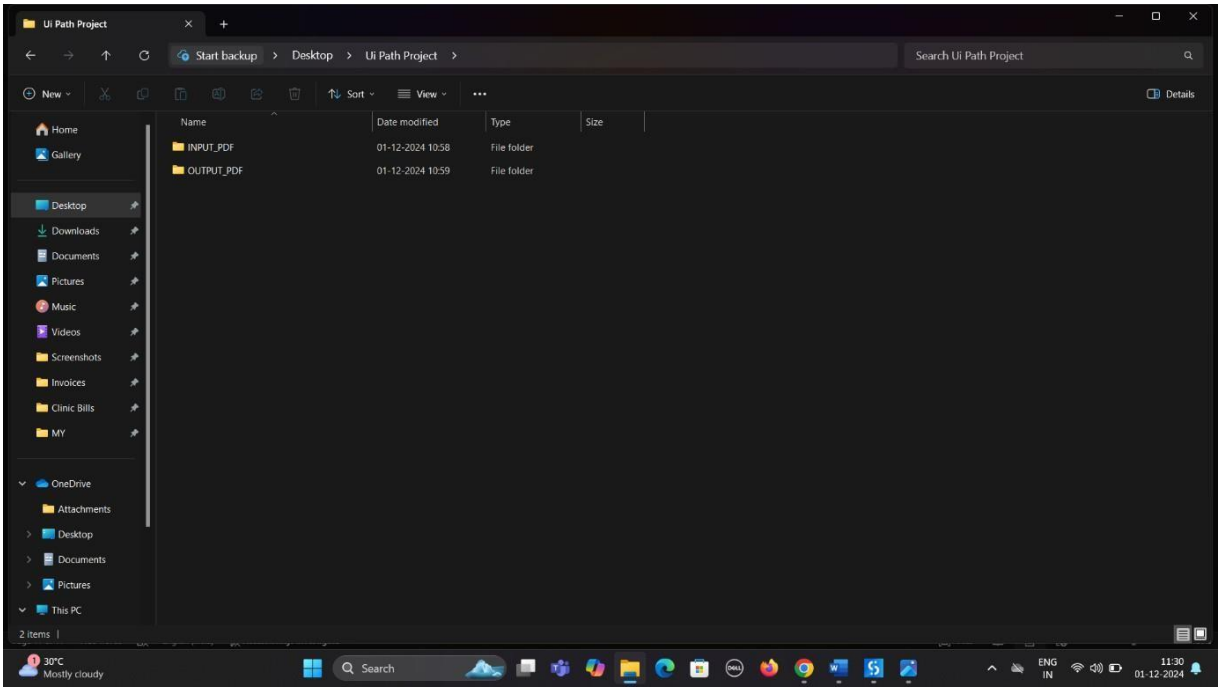


Fig 1: Create Folder for Input and Ouput

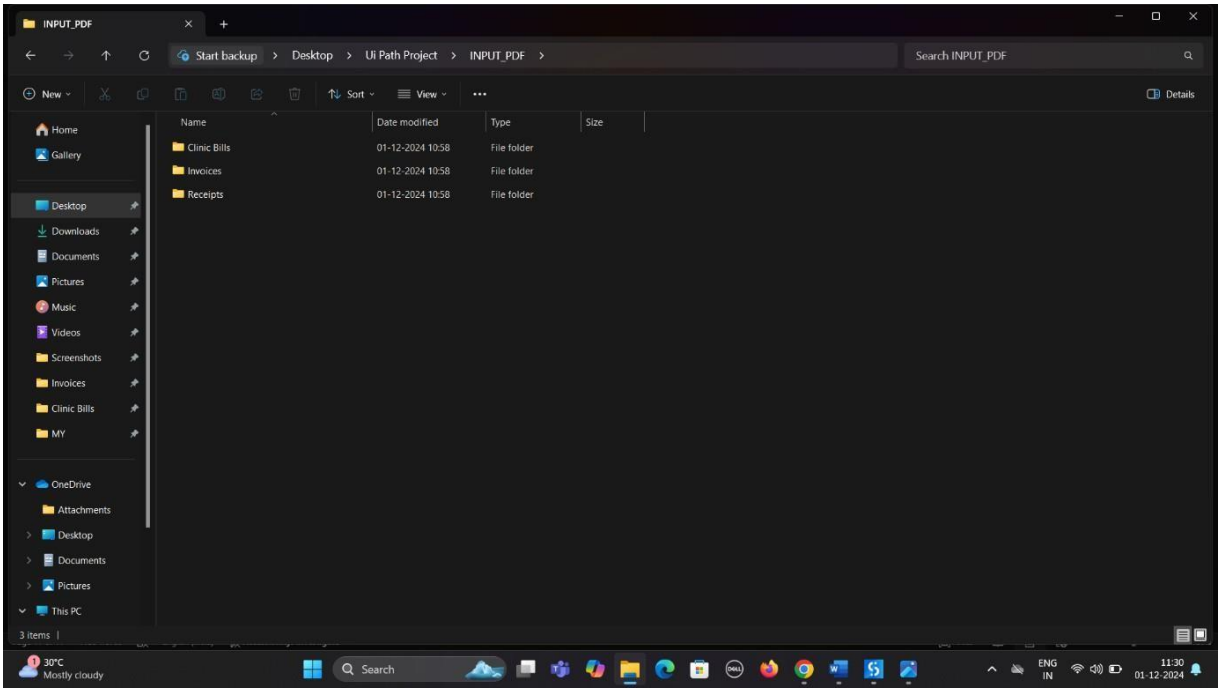


Fig 2: Folder for INPUT_PDF Save PDF for one of three folders—**Invoices**, **Receipts**, or **Clinic Bills**

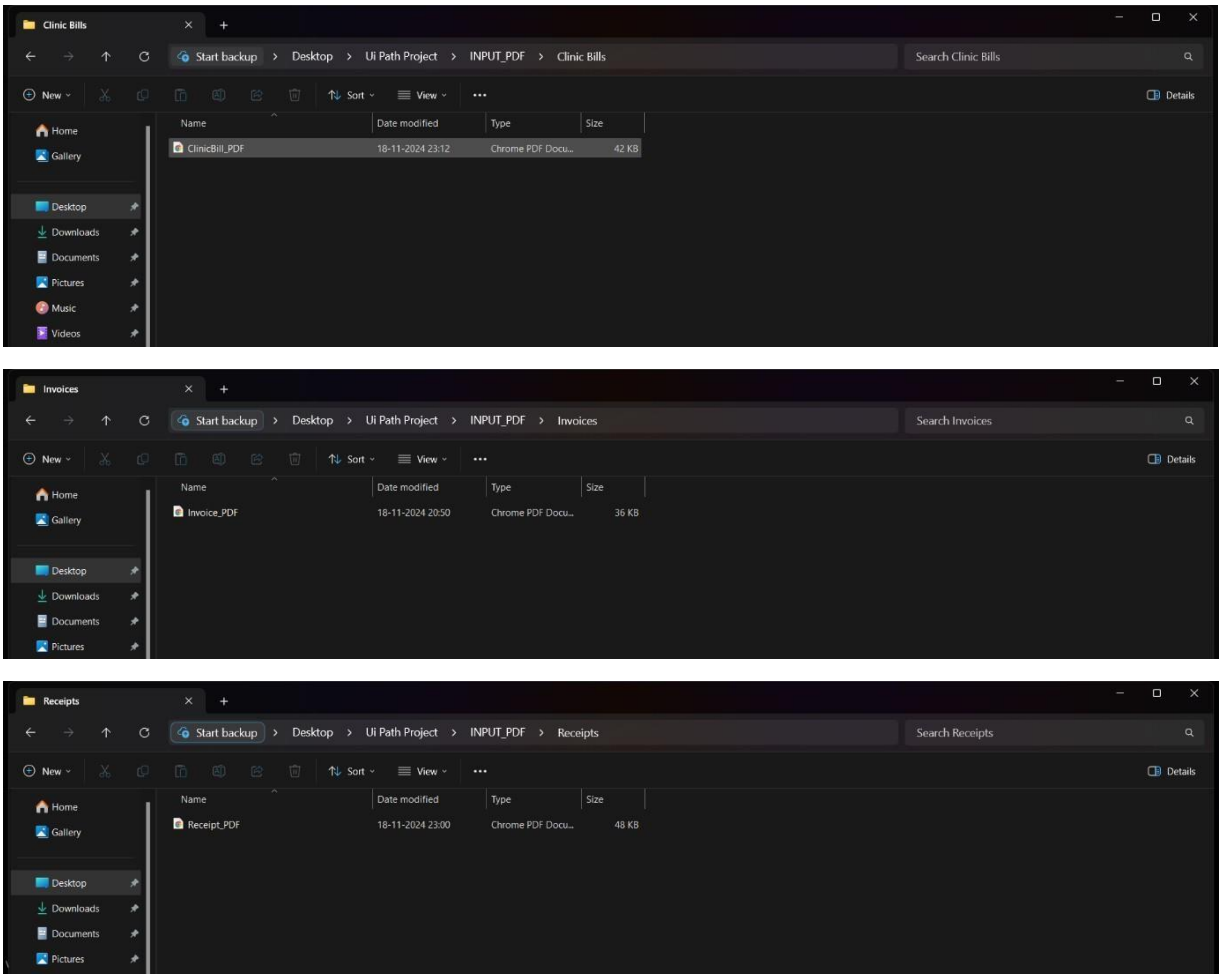


Fig 2.1: Store PDF file inside the each require folder, each PDF Contains 3 pages

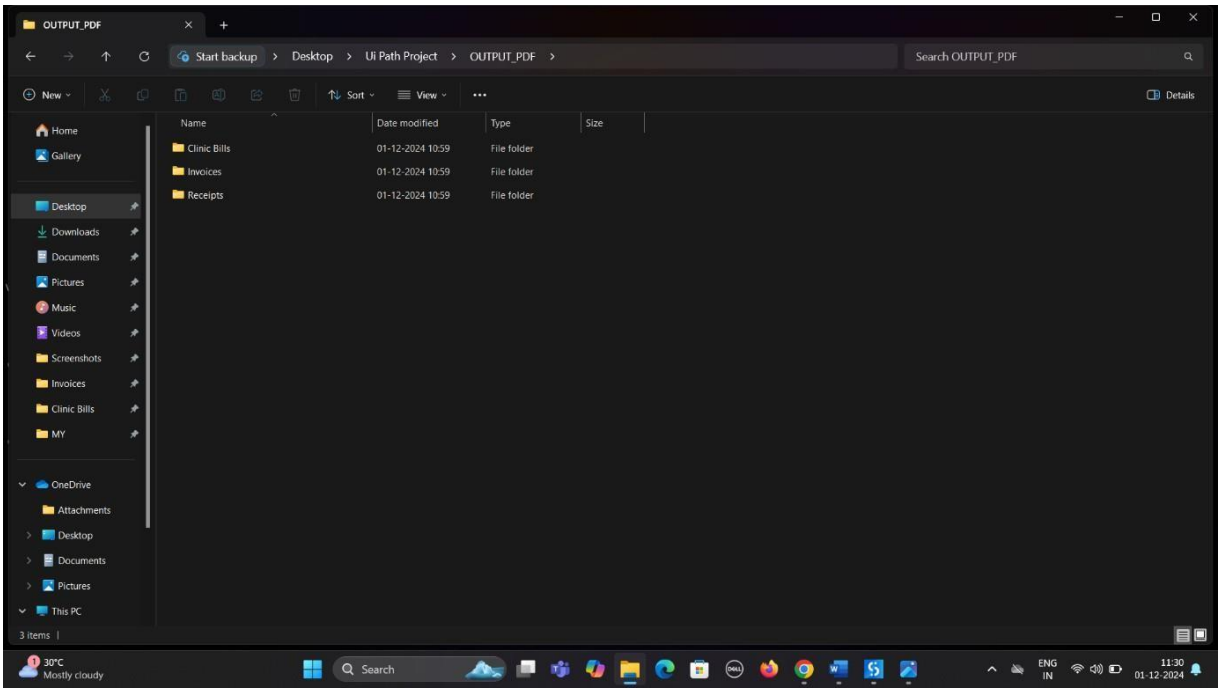


Fig 3. Folder for OUTPUT_PDF

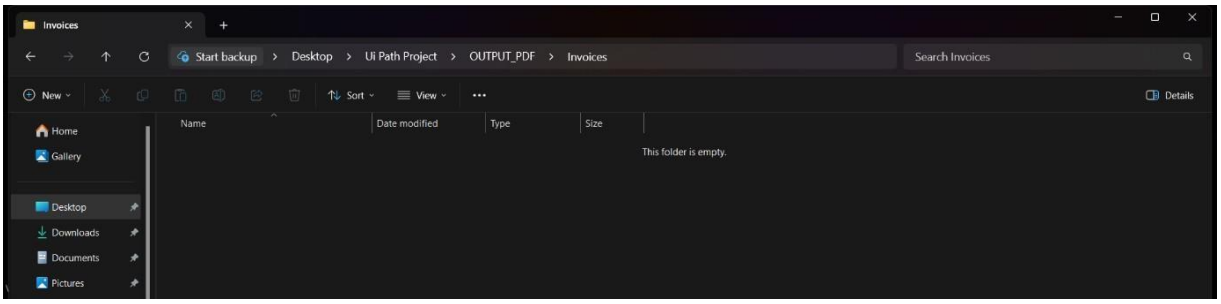


Fig 3.1: Initially the folder was empty

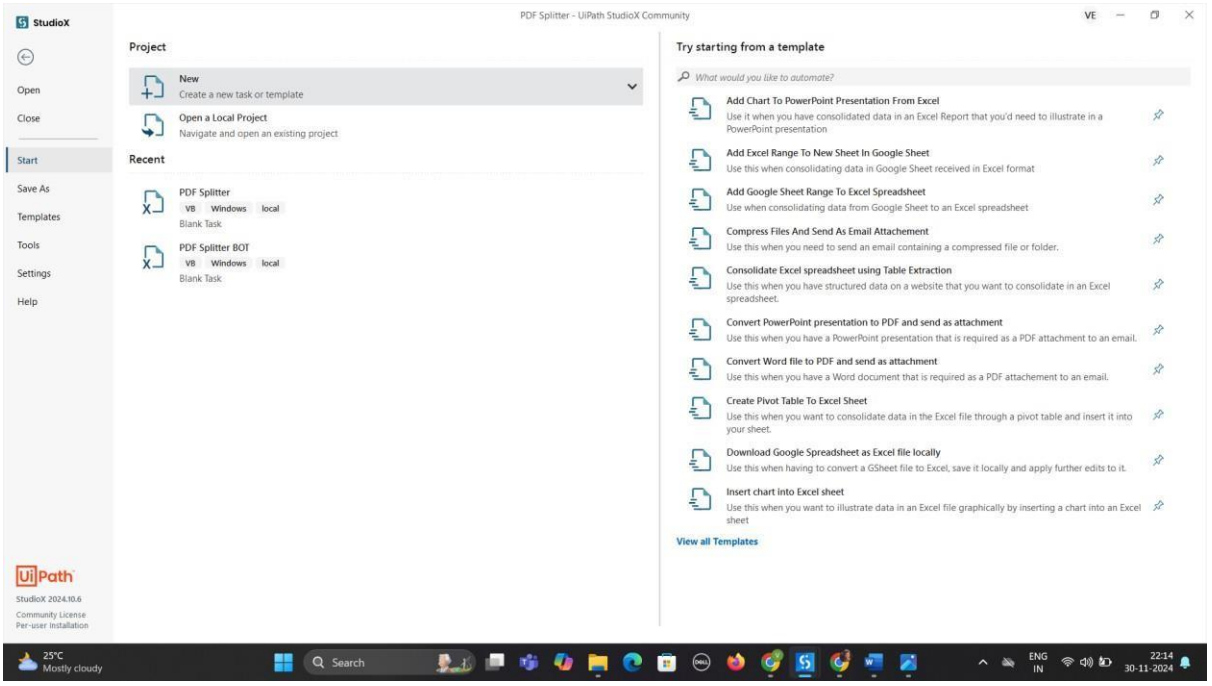
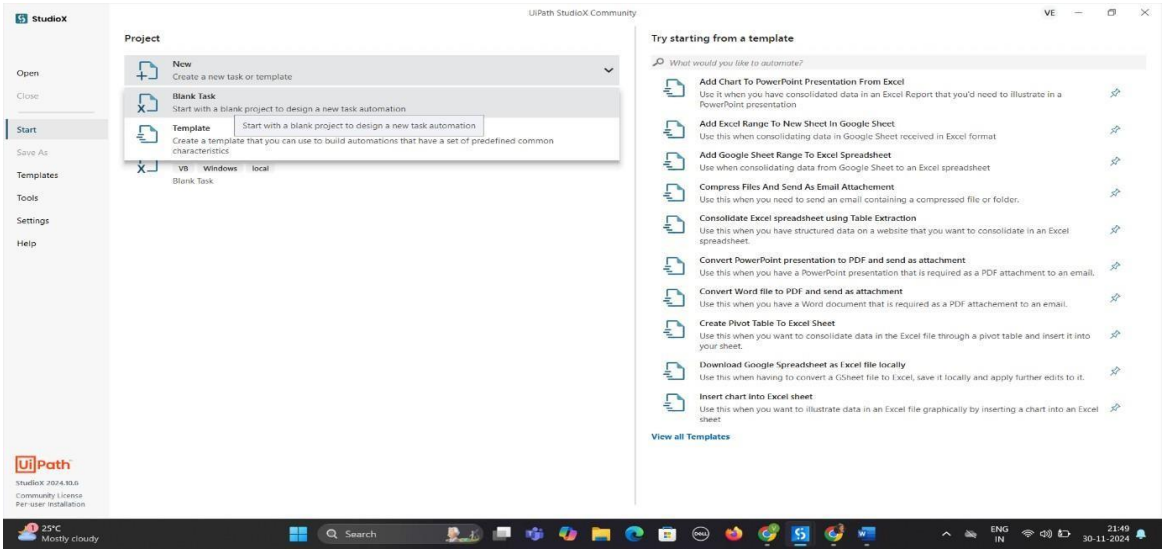


Fig 4: Install UiPath Studio and Open



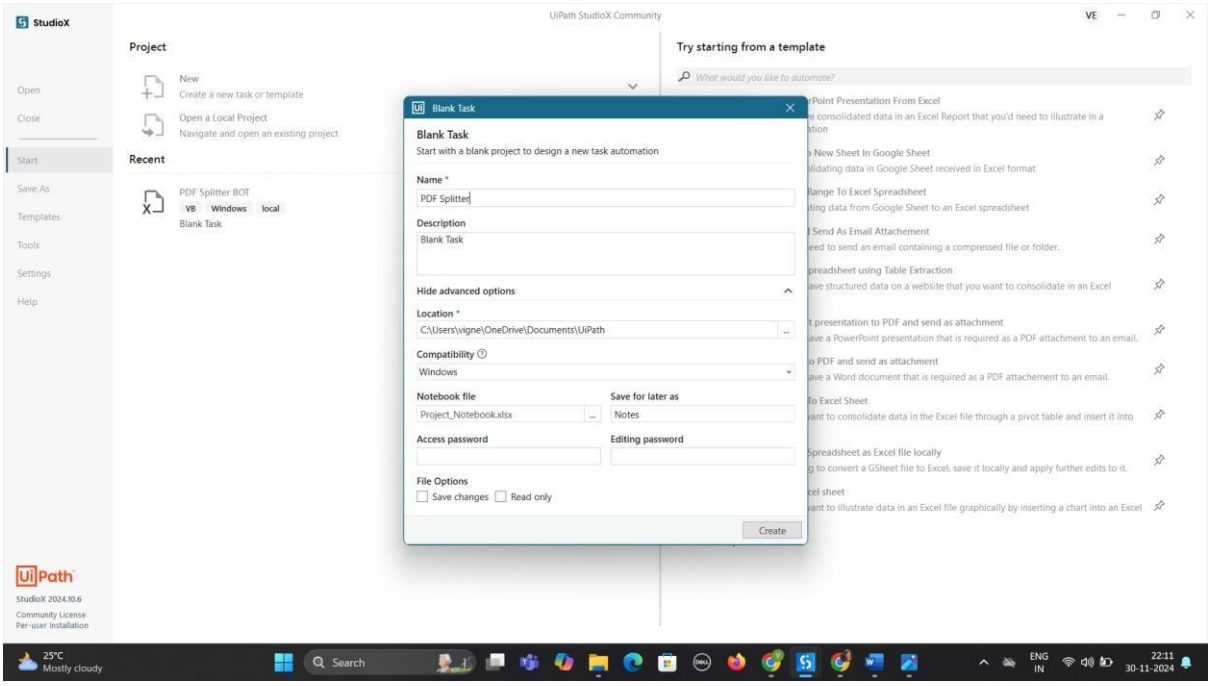
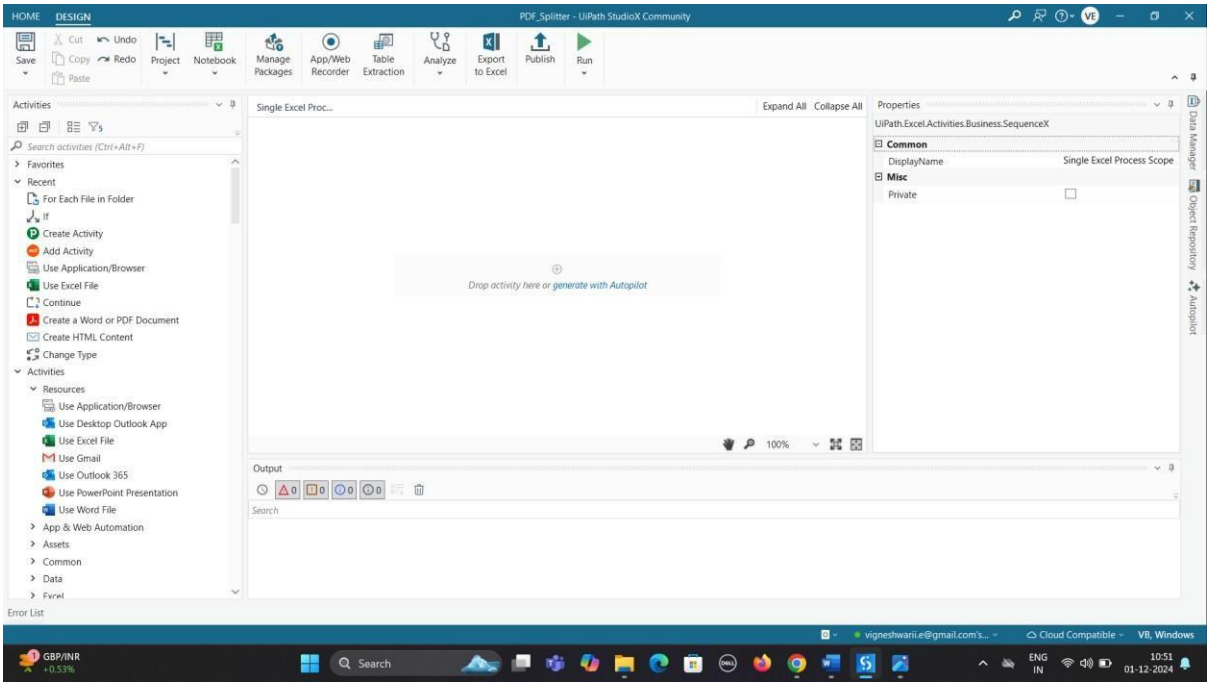


Fig 4.1: Click **"New Process"** -> Project Name: **PDF Splitter BOT** -> Click **"Create"**



4.2: New project will open

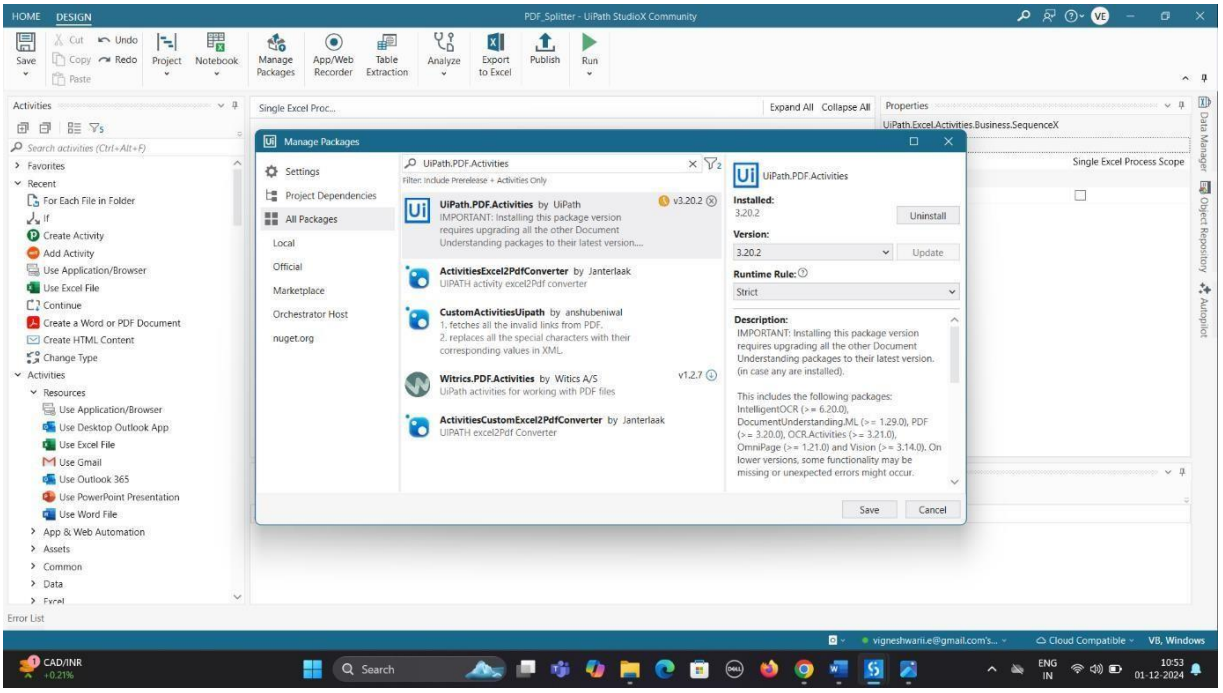


Fig 5: Go to "**Manage Packages**" and Search for PDF Activities then Save and Close

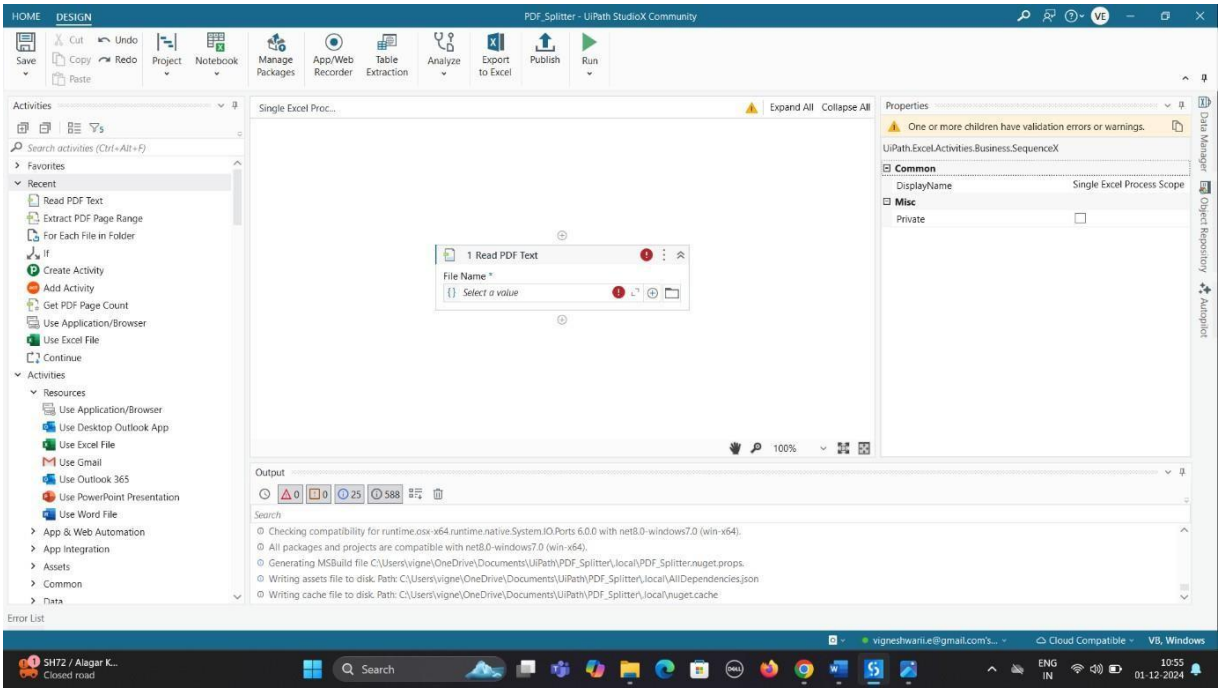
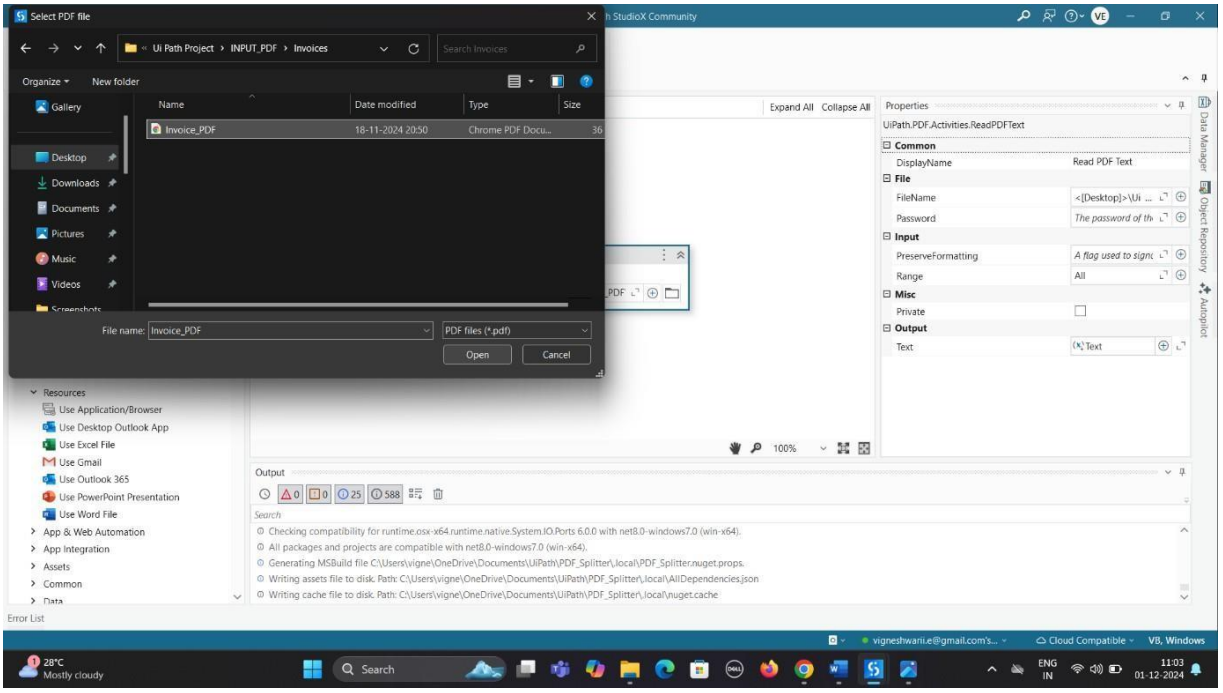


Fig 6: Drag and Drop the Read PDF Text from Activities



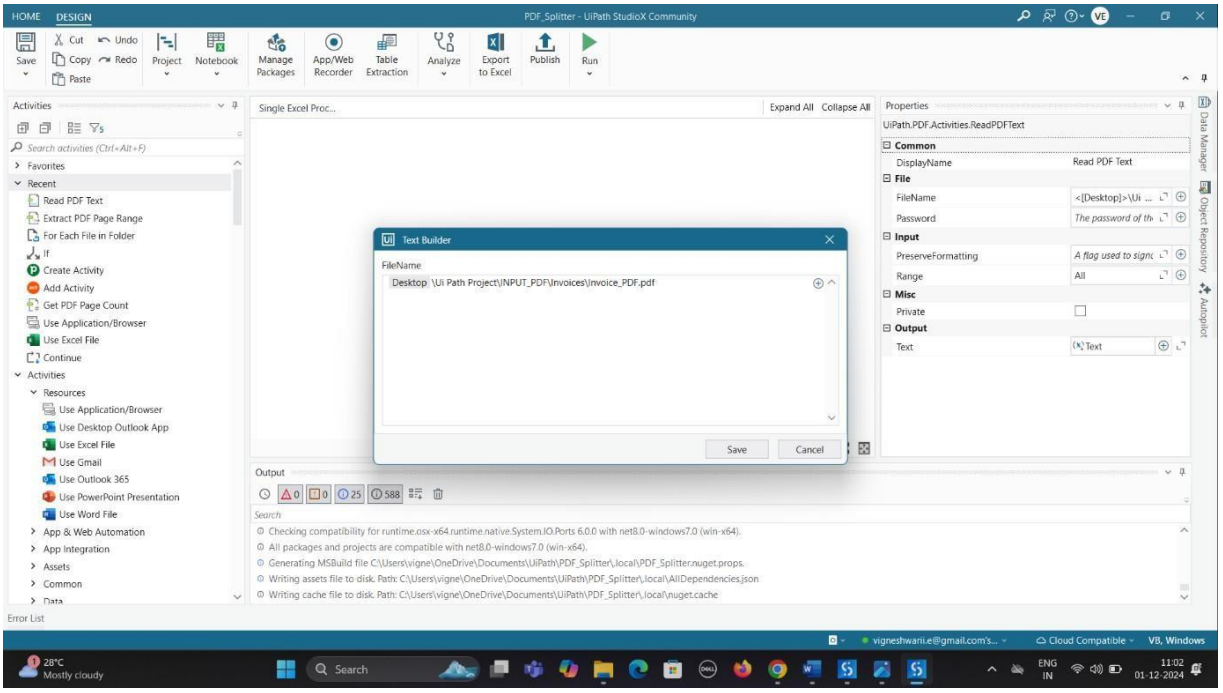


Fig 7: Set path for INPUT_PDF

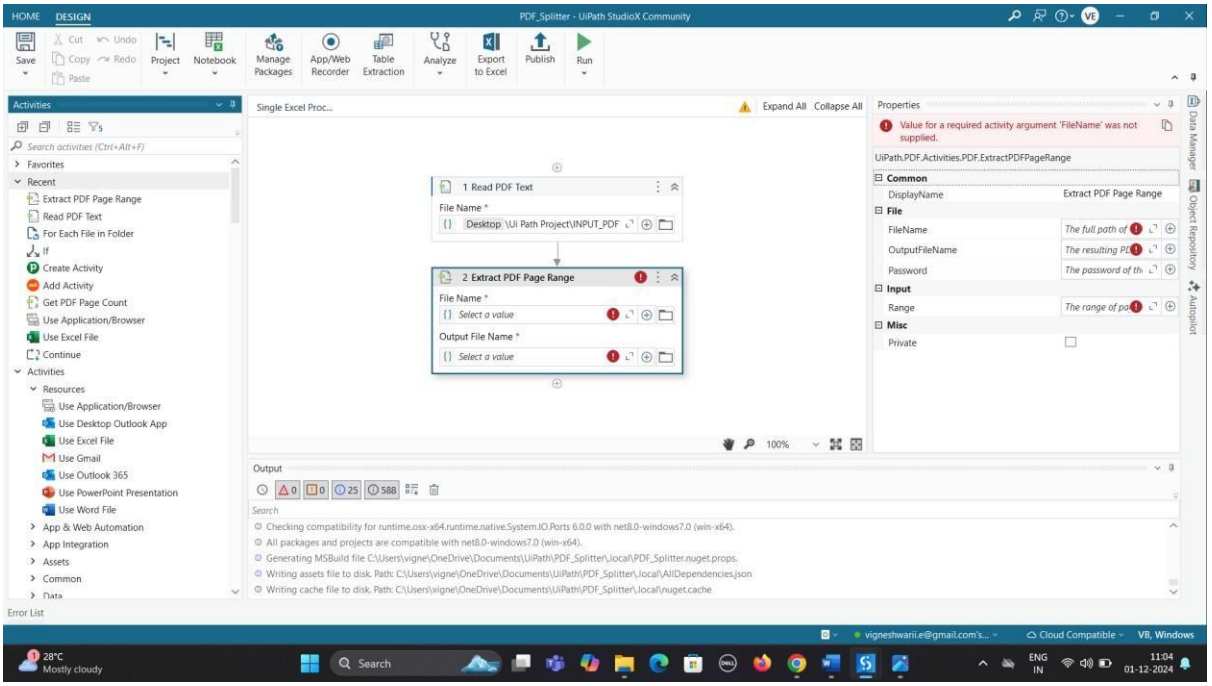


Fig 8: Drag and Drop the Extract PDF Page Range

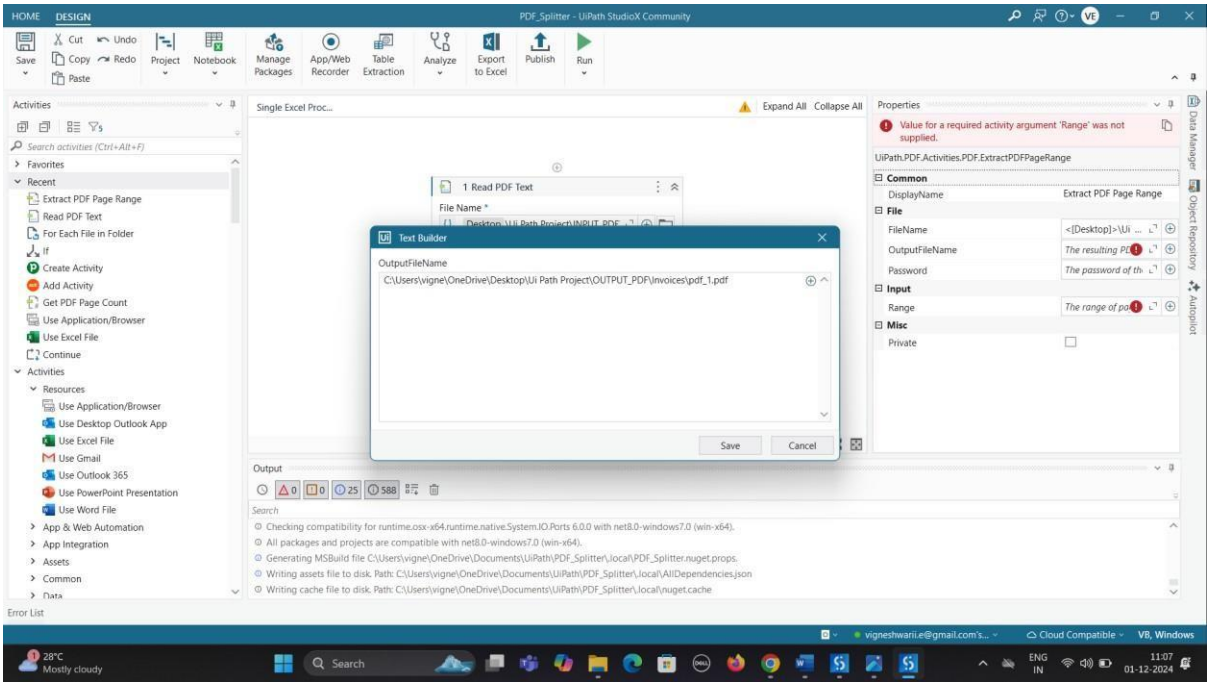


Fig 8.1: Set path for INPUT_PDF

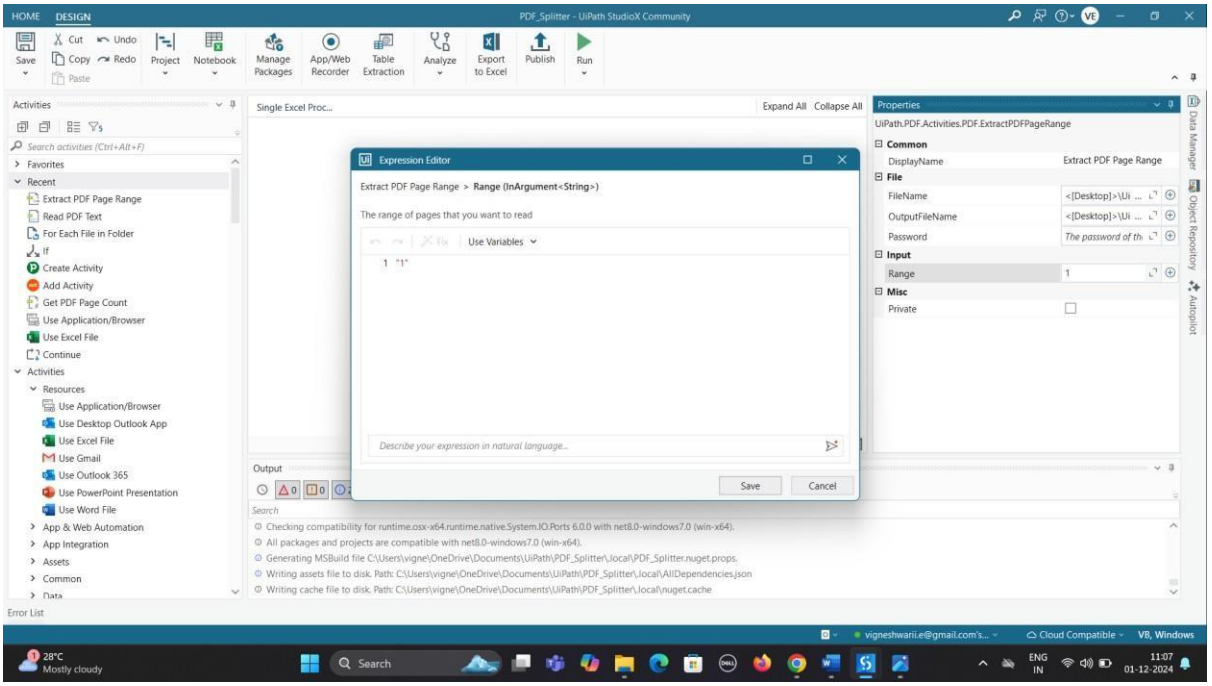


Fig 8.2: Set Range to split the document into individual pages like “1”, “1-2”

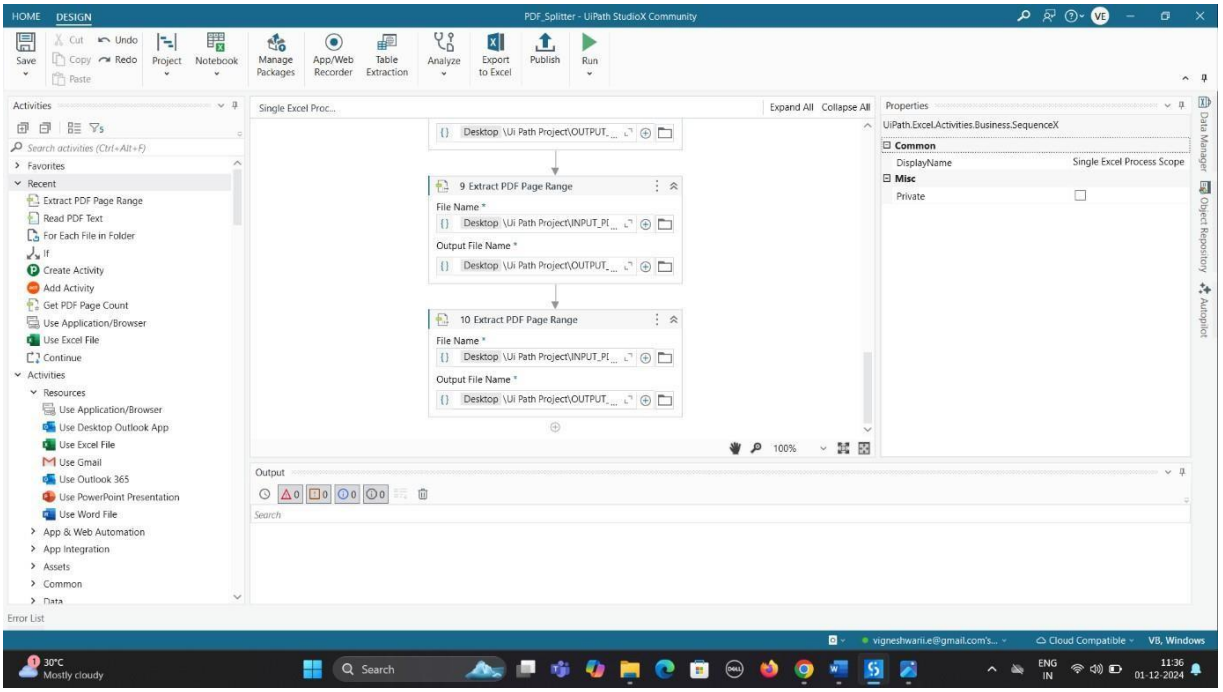


Fig 8.3: Test the Workflow

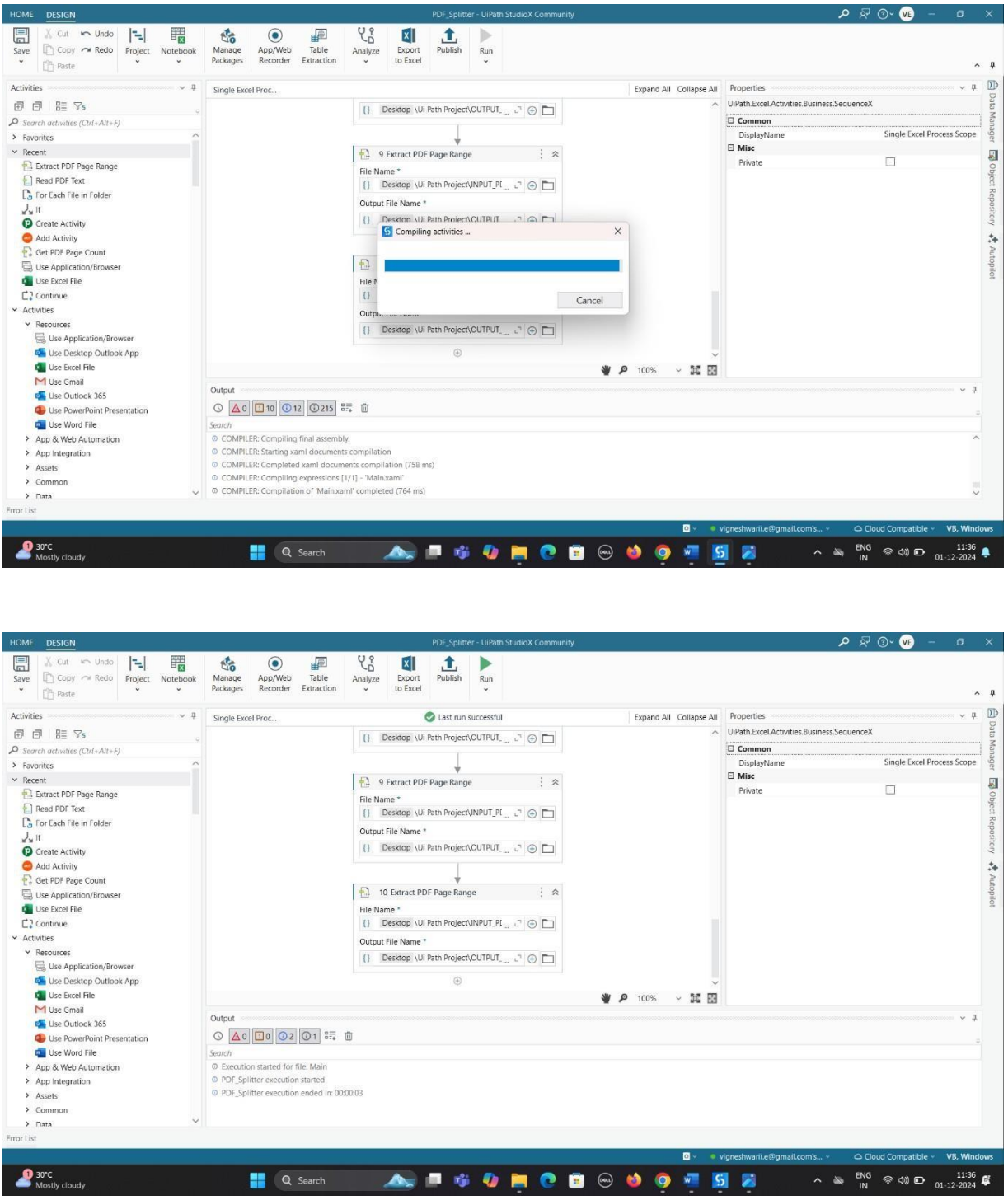


Fig 8.4: Click the **"Run"** button in UiPath Studio to test your bot

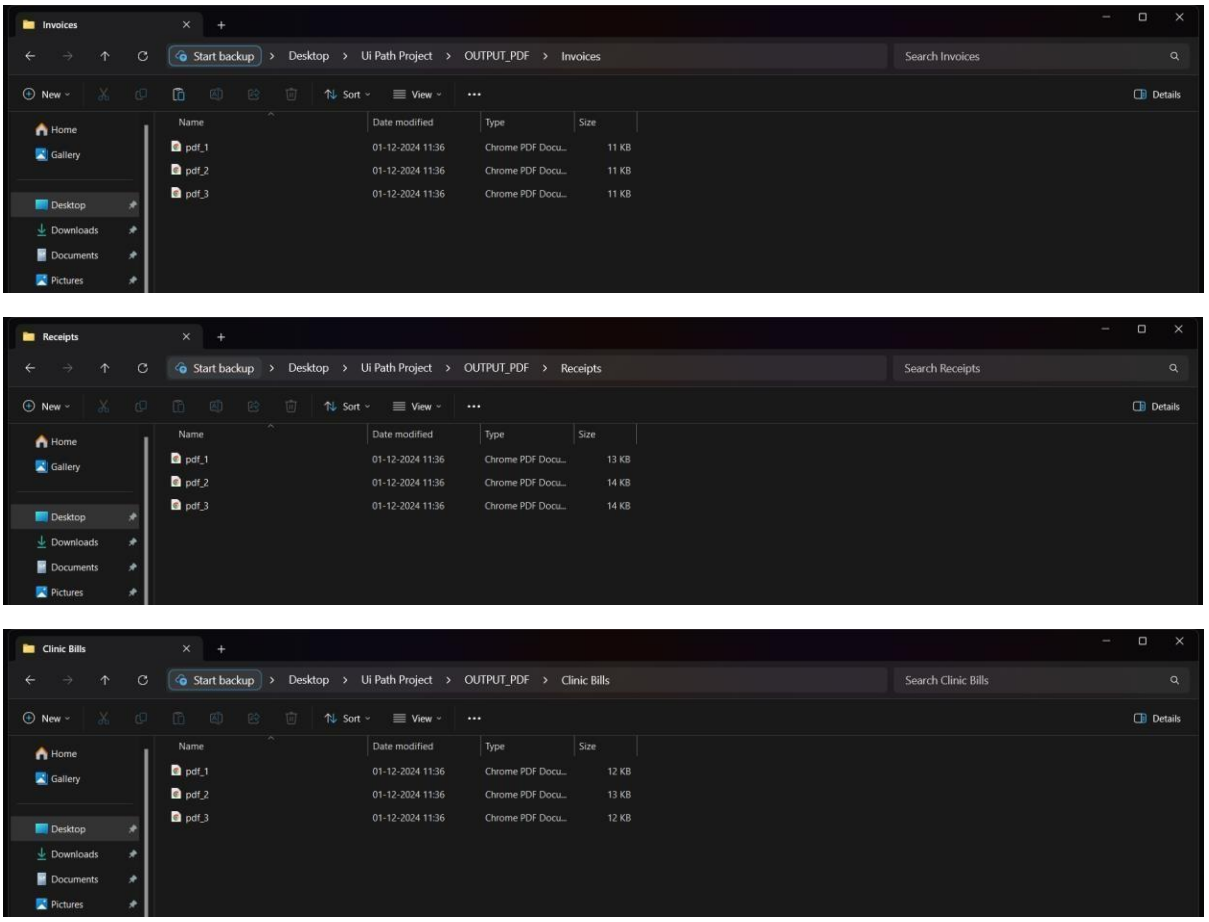


Fig 8.5: Check if the files are saved in the correct folders

Conclusion:

The **PDF Splitter BOT** project successfully demonstrates the potential of robotic process automation (RPA) in simplifying and optimizing document management tasks. By leveraging the advanced capabilities of UiPath, the bot efficiently automates the tedious process of reading, classifying, splitting, and saving PDF files based on their content. This solution provides businesses with a reliable and error-free way to handle large volumes of documents such as invoices, receipts, and clinic bills.

The project has significantly reduced the manual effort required for sorting and organizing PDF documents, resulting in improved efficiency and accuracy. The bot's ability to handle both text-based and image-based PDFs ensures its versatility and applicability in various scenarios. Additionally, the modular design of the workflow allows for easy customization and scalability, enabling the bot to adapt to future requirements.

This project not only showcases the practical benefits of automation in the workplace but also highlights how RPA can free up valuable human resources for more strategic tasks. By adopting solutions like the **PDF Splitter BOT**, organizations can achieve better productivity, minimize errors, and maintain a more organized document management system.

Looking ahead, enhancements such as advanced machine learning models for document classification, better OCR integration for multi-language support, and cloud storage capabilities can make the bot even more powerful. The success of this project underscores the transformative potential of RPA in addressing real-world business challenges.