

**Technologies for Financial Industry**

Year 2/3 (2022/23), Semester 3/5

SCHOOL OF INFOCOMM TECHNOLOGY

Diploma in Financial Informatics

Diploma in Information Technology

**ASSIGNMENT**

**STUDENT’S PARTICULARS**

|  |  |
| --- | --- |
| **Student ID** | **Name** |
| S10223630 | Armirola Roseller III Tumolva |

Youtube URL: <https://youtu.be/FX9CjJvagZs>

Table of Contents

[Overview 3](#_Toc110543661)

[Data Extraction – Financial Data 3](#_Toc110543662)

[Data Extraction – SDG Data 9](#_Toc110543663)

[Data Handling – Financial Data 15](#_Toc110543664)

[Data Handling – SDG Data 22](#_Toc110543665)

[Data Analysis & Comparison 24](#_Toc110543666)

[PDF Automation 26](#_Toc110543667)

[Email Automation 34](#_Toc110543668)

[Reflections 37](#_Toc110543669)

[UiPath Certification 38](#_Toc110543670)

[References 39](#_Toc110543671)

[Appendices 40](#_Toc110543672)

# Overview

This individual report covers the Robotic Process Automation (RPA) solution that I have made to help the team choose a company to invest in – in this case, Meta Platforms (Facebook).

Graphical user interface, text, application

Description automatically generated

Figure . Main Workflow

# Data Extraction – Financial Data

I started by creating a workflow that extracts financial data pertaining to Meta Platforms’ Finance Matrices from 2 websites – Financial Modeling Prep and CSIMarket ([see Appendix A – Websites Used](#_Appendices))

I will first create some pre-defined variables:

* **financialRatiosFileName** is the name of the file when downloading the Financial Ratios Excel Sheet from Financial Modeling Prep website
* **incomeStatementFileName** is the name of the file when downloading the Income Statement Excel Sheet from Financial Modeling Prep website
* **savingFolderPath** is the full folder path to the Workfiles folder of this process
* **dateTime** is the current date

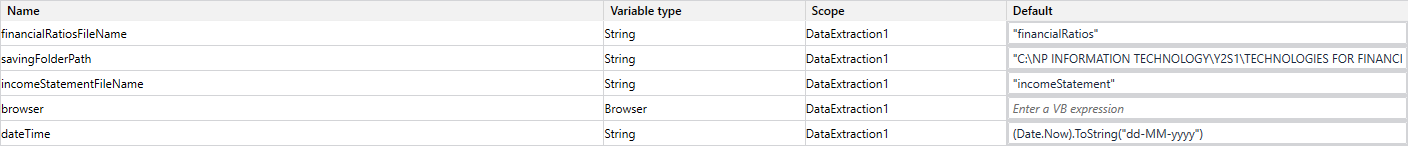


Figure . Pre-Defined Variables

Then, I used a **“Open Browser” activity** to open the Financial Modeling Prep website in **Chrome (BrowserType.Chrome)** and used **Basic Recording** to search for the desired stock “Meta Platforms” and download the Financial Ratios Excel sheet. This crosses out one of the required RPA solution techniques – any 1 of the 4 available recording tools.

Graphical user interface, application

Description automatically generated

Figure . Download Financial Ratios Excel Sheet

To make sure that the UI actions are dynamic, I had to configure the **selectors** to take in account for any situation, like changes to the site name and html tags. This would satisfy half of the requirement – at least 2 types of control features such as selectors, web elements, anchor base, etc.

Figure 4 shows an example of the selectors I had to configure for this **Basic Recording** Sequence.

Graphical user interface, text, application

Description automatically generated

Figure . Search - Meta Platforms Selectors

Upon clicking the download button for the financial ratios excel sheet, I checked to see whether the user running the solution had enabled the “Save As” download option for their Chrome browser, as it is required to download the file in the specific Workfiles folder.

If the “Save As” window exists, it downloads in the savingFolderPath along with the financialRatiosFileName and today’s date. Success will be logged.

Otherwise, output a fatal log level message and **throw a new exception**. With this, it satisfies the requirement of at least 2 types of control features (used Selectors, Error Handling and If-Else Control)

Graphical user interface

Description automatically generated

Figure . Check "Save As" Exists

The same thing will happen but for downloading the Income Statement Excel Sheet.

Graphical user interface

Description automatically generated

Figure . Download Income Statement Excel Sheet

I noticed that the downloaded financial ratios excel sheet **did not** have the Return on Investment (ROI) metric which my team required to compare the companies to invest in later. Thus, I opened a new chrome browser going to **CSIMarket** website for Meta Platform’s **annual ROI data**.

I performed Data Scraping to get the data on the website and extracted the data into a DataTable.

Graphical user interface, application

Description automatically generated

Figure . Retrieve ROI Metric

With the retrieved DataTable, I wrote the data into an excel sheet to combine with the financial ratios excel sheet later.

This would finish the retrieval of the necessary financial data from Meta Platforms.

Graphical user interface

Description automatically generated

Figure . Writing ROI DataTable to Excel Sheet

After retrieving the financial data, I moved on to retrieve the data pertaining to the chosen genre of the assignment, Sustainability Development Goal #9 – Industry, Innovation and Infrastructure. This concludes the [DataExtraction1 workflow](#_Data_Extraction_–_1).

# Data Extraction – SDG Data

For extracting the SDG data, I created a workflow that extracts data pertaining to Meta Platforms’ ESG rating and sustainability development progress statement from Meta Platform Facebook’s investor webpage and sustainalytics websites.

I need to pass in some arguments to update the **esgRating** variable to pass it to other workflows.

Graphical user interface, application

Description automatically generated

Figure . Pass in arguments

I will then first pre-define some variables:

* **savingFolderPath** is the full path for the workfiles folder
* **sdgGoalFileName** is the specified name for the downloaded PDF file
* **dateTime** is the current date

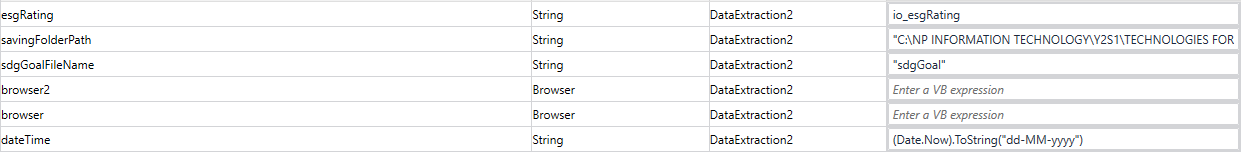




Figure . Pre-define variables

After that, I opened a new chrome browser directing to the Meta Platforms investor webpage. Then I use a **UI Automation -> Element -> Mouse -> Click activity** to select the 2021 sustainability report. To make this selector more dynamic for the future years’ sustainability report, I configured the selector as such by changing the “name” attribute to change the 2021 in “2021 Sustainability Report” to “\*Sustainability Report” using the asterisk (\*) to accept any value before “Sustainability Report”. Then, I downloaded the PDF file using the **UI Automation -> Image -> Mouse -> Click Image activity**.

Graphical user interface, application

Description automatically generated

Figure . Download 2021 Sustainability Report

Graphical user interface, text, application, email

Description automatically generated

Figure . Configure dynamic selectors

Similar to the [DataExtraction1 workflow](#_Data_Extraction_–_1), I set up a Save As checker to see if the user has enabled the “Save As” option in their chrome browser and manually save to the desired location and file name.

Graphical user interface

Description automatically generated

Figure . Check if "Save As" is enabled in chrome

After downloading the sustainability development report for 2021, log the progress and use the **Close Application activity** and selecting the close button for the chrome browser instead of the **Close Tab activity**. This is because there will be two tabs that will pop-up up to this stage in the workflow.

Graphical user interface, text, application

Description automatically generated

Figure . Close chrome browser application

Next, we want to extract the data pertaining to the company’s ESG Risk Rating. First we open a new chrome browser to the sustainalytics website.

In the website shows the risk rating and in order to retrieve it, I used the **Get OCR Text** **activity** along with the **Screen Scraping tool**. I will output the text and update the **esgRating** variable to be passed to other workflows to use

Graphical user interface

Description automatically generated

Figure . Screen scrape with OCR ESG Risk Rating

I used **Microsoft OCR** and configure the properties as shown:

* English (United States) language
* Screen profile
* Scale of 2

Table

Description automatically generated

Figure . Microsoft OCR properties

And this concludes the [DataExtraction2 workflow](#_Data_Handling).

# Data Handling – Financial Data

For handling the Financial Data, I created a workflow that formats the financial data excel sheets acquired from the [DataExtraction1 workflow](#_Data_Extraction_–_1). This workflow satisfies the requirement to use excel and data table activities for the tabulation of statistical components.

First, I pre-defined a set of variables:

* **dateTime** is the current date
* **savingFolderPath** is the full path for the workfiles folder
* **financialRatiosFileName** is the specific file name for the financial ratios excel sheet
* **incomeStatementFileName** is the specific file name for the income statement excel sheet
* **returnOnInvestmentFileName** is the specific file name for the ROI excel sheet

Graphical user interface, chart

Description automatically generated with medium confidence



Figure . Pre-define variables

I then read 2 excel files that I will need to filter and combine – financialRatios and returnOnInvestment.

Graphical user interface, text, application

Description automatically generated

Figure . Read financial ratios and ROI excel sheet files

After reading the excel sheets into data tables, I first changed the year column to an object datatype using a non-standard package activity from BalaReva.EasyDataTable.Activities **Change Column DataType** **activity** which satisfies the assignment requirement to use non-standard UiPath packages.

After converting the the year column’s data type, I filtered the financial ratios data table to only contain the 5 years that me and my team wants (2017-2021) and the respective metrics that we are going to use to compare our companies’ finances with (P/E, P/B, ROI, ROE, ROA and NPM). So using the **Filter Data Table activity**, I will keep rows with the desired year values and keep columns with the desired ratio metrics.

Graphical user interface, text, application, email

Description automatically generated

Figure . Filter financial ratios data table

Table

Description automatically generated

Figure . Filter Data Table rows

Graphical user interface, text, application, email

Description automatically generated

Figure . Filter Data Table columns

Then, I proceeded to filter the return-on-investment data table as I only want the row that contains the total annual return-on-investment values as highlighted in the excel sheet

So, using the **Filter Data Table activity**, I kept the row that contains the desired values.

Table

Description automatically generated



Figure . Return on investment excel sheet (highlighted desired row)

Graphical user interface, text, application, email

Description automatically generated

Figure . Filter ROI data table

Graphical user interface, text, application, Word

Description automatically generated

Figure . Filter Data Table rows

After filtering the 2 data tables, I proceeded to integrate the values from the return-on-investment data table to the financial ratios data table.

First, I added a new ROI column with the **Add Data Column activity**, then renamed all the column headers using the **Multiple Assign activity** as the original financial ratios excel sheet had a column header naming convention that I don’t like.



Figure . Financial ratios excel sheets column headers

Graphical user interface, table

Description automatically generated

Figure . Rename column header names



Figure . C# code

Then I had to put in the ROI values 1 by 1 using a **For Each Row in Data Table activity** which reiterates the financial ratios data table and insert the values of the data cells in the row from the return-on-investment data table.

Also, because the year values were in full date time format (e.g., 12/31/2021 00:00:00), I had to rename the year values to their respective year (i.e., 2021).

Graphical user interface

Description automatically generated

Figure . Loop to add values from ROI data table to financial ratios data table

To directly compare with the other companies from the other team’s finances, I had to compute the averages for the various financial ratio metrics. To do this, I had to use some C# to find the average as shown below.

Graphical user interface, application

Description automatically generated

Figure . Multiple Assign calculated average for all metrics

Graphical user interface, text, application

Description automatically generated

Figure . C# code

Then, I will add the averages of all the metrics into a new row using the **Add Data Row activity** to the **financialRatios\_dt**.

Graphical user interface, text, application, email

Description automatically generated

Figure . Insert average row

After I finished editing the financial ratios datatable, I exported it into a new excel sheet file (.xlsx) to the specified workfiles folder and file name using the **Excel Application Scope** and **Write Range** activities to use it for the [DataAnalysis workflow](#_Data_Analysis_&) to create a column chart for visualization. I will check the AddHeaders property as well.

I noticed that the year values will look weird (13/7/1905 00:00:00) because the format of the cells are not set to “Text”, thus I used a non-standard package from **BalaReva.Excel.Activities**, , which results in the appropriate year value (e.g., 2021). This satisfies the requirement to use additional non-standard packages to enhance the automation.

Graphical user interface, application

Description automatically generated

Figure . Write finished financial ratios data table to excel sheet

The resulting excel sheet from this workflow will look as such:

A screenshot of a computer

Description automatically generated with medium confidence

Figure . Resulting excel sheet

And this concludes the [DataHandling1 workflow](#_Data_Handling_–_1).

# Data Handling – SDG Data

For handling the SDG Data, I created a workflow that formats the SDG Goal 9 progress statement and the company’s ESG Risk Rating acquired and stored in the variables – **sdgGoal** and **esgRating** respectively.

I had to import 2 in-out arguments which will update the **esgRating** and **sdgGoal** variables in the main workflow.

Graphical user interface, application

Description automatically generated

Figure . Import arguments

First, I pre-defined the variables to be updated for the workflows later:

* **esgRating** and **sdgGoal** are the activities to be updated in this workflow
* **dateTime** is the current date

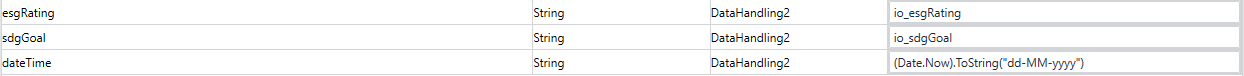


Figure . Pre-define variables

I added a **Read PDF Text activity** to read the downloaded PDF that contains the company’s sustainability report for 2021. I specified the folder the file is in as well as the specified to read only page 18 and output the text to the **sdgGoal** variable

Graphical user interface, text, application

Description automatically generated

Graphical user interface, table

Description automatically generated

Figure . Read PDF Text activity properties

Then, I assigned the formatted text from the **sdgGoal** variable because the text from the PDF is not in the standard format as shown:

Graphical user interface

Description automatically generated

**Graphical user interface, text, application, Word

Description automatically generated**

Figure . Substring the sdgGoal text

After that, log the message of the completion of the [DataHandling2 workflow](#_Data_Handling_–)

# Data Analysis & Comparison

For data analysis, I created a workflow that creates a column chart with the data in the excel sheet we have exported from the [DataHandling2 workflow](#_Data_Handling).

First, I have to specify 2 pre-defined variables:

* **dateTime** is today’s date
* **savingFolderPath** will be the full path for the Workfiles folder

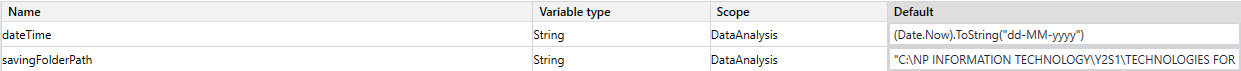
****

Figure . Pre-define variables

To create this column chart, I used a non-standard package – **BalaReva.Excel.Activities** – that contains a **Column Chart activity** which I will be using.

Graphical user interface, text, application

Description automatically generated

Figure . DataAnalysis workflow

Let us take a look at the necessary properties I have set for this activity.

First, I applied the cell range to read all of the cells in the resulting excel sheet (“A1:G7”). I specified the title of the chart to be “Column Chart” and also specified the full path of the **email attachments folder** to save a copy of the image there for the email. For specific chart options, I set is as specified in the figure below. Next, for the input file path, I used the pre-defined variables to read the resulting excel sheet and also specified the sheet name accordingly.

Graphical user interface, application

Description automatically generated

Figure . Column chart activity properties

Then, log the completion of the DataAnalysis workflow.

The resulting chart image will look as such:

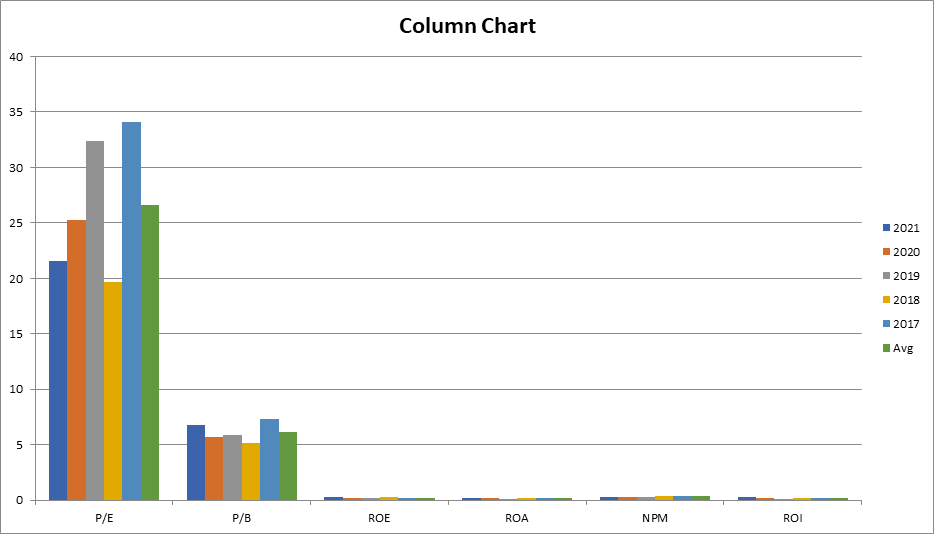


Figure . Resulting column chart image

We can already observe the decrease of the company’s P/E ratio from 2019 to 2021 most definitely due to the COVID-19 pandemic. We can also see that there are no negative values for the past 5 years for all metrics, indicating that the company has managed to hold well despite the pandemic.

This would therefore conclude the [DataAnalysis workflow](#_Data_Analysis_&).

# PDF Automation

I created a workflow to automate the editing of the PDF file with the chosen 1 of the 4 templates to be sent to my teammates via email.

This workflow will import 3 arguments:

* **io\_templateChoice** which will be updated in this workflow
* **io\_sdgGoal** and **io\_esgRating** which is updated from the [DataHandling2 workflow](#_Data_Extraction_–)

Table

Description automatically generated

Figure . Import arguments

First, I set a few pre-defined variables:

* **dateTime** is the current date
* **savingFolderPath** and **savingFolderPath2** is the specific full paths for the email attachments and workfiles folder
* **confirmationChoice** is pre-set to “No” so the while loop will keep going even if the user has not chosen a template and/or confirmed their choice
* **replace1,2,3,4** is what I will replace the text from the PDF template (this is separated into multiple variables as there is a limit to how many characters you can replace with one **Replace Text in Document activity**
* **ticker**, **companyProfile**, contains the investment ticker of Meta Platforms as well as a brief introduction of the company
* **sdgGoal** and **esgRating** is acquired from the [DataHandling2 workflow](#_Data_Handling_–)

Graphical user interface, table

Description automatically generated



Figure . Pre-define variables

So I will start the workflow with a **While loop** that keeps iterating the template choosing process until the user is satisfied with the template they wish to use.

A picture containing graphical user interface

Description automatically generated

Figure . While loop

Then, I added in an **Input Dialog activity** to prompt the user for which template number (1-4) they wish to use with multiple choice input type.

Graphical user interface, text, application, email

Description automatically generated

Figure . Input Dialog Prompt for Template Choice

Then, I set up **nested if-else activities** to check what template number the user has chosen and accordingly allocate the order to the **template** variable

Graphical user interface

Description automatically generated

Figure . Nested If-Else activities

I added another **Input Dialog activity** to confirm with the user if they wish to proceed with the PDF Automation with the selected template, if they choose “No”, the process up to this activity will repeat again. Otherwise, the process will continue.

Graphical user interface, text, application, email

Description automatically generated

Figure . Input Dialog Confirmation

After choosing the PDF template, I continued to use **PDF and Word activities** to read the pdf text and save it to the **text** variable, set the PreserveFormatting property to True and read all pages. This satisfies one of the required RPA solution techniques where I am supposed to use **PDF Automation** with data extraction from the attached PDF documents (4 varying formats).

Graphical user interface, application, Teams

Description automatically generated

Figure . Read PDF template

Graphical user interface, table

Description automatically generated

Figure . Read PDF Text activity properties

Then, with the extracted text, I appended it to a new Word document using the **Word Application Scope** and the **Append Text activities** with the saving folder path to the workfiles folder and the name as the number of the template chosen with the current date. I also want to check the property to create a new folder if the word document does not exist

Graphical user interface, text, application, email

Description automatically generated

Figure . Append text from PDF document

Then, I will add another **Word Application Scope activity** that carries out the **Replace Text in Document** activities to replace the template’s text with the information acquired before.

Graphical user interface, application

Description automatically generated

Figure . Replace text

Inputting the ESG Risk Rating and the SDG Goal 9 Progress statement might have an error that can occur because the **Replace Text in Document** activity does not allow replacement with NULL values which would be triggered because the variables **esgRating** and **sdgGoal** were not updated. Thus, I had to encase these activities in a **Try Catch activity** to work around it the problem.

Graphical user interface, application, email

Description automatically generated

Figure . Try Catch activities

If an error occurs, the activity will catch a general exception and log a warning. The text to be replaced will be a blank string instead.

Graphical user interface, application

Description automatically generated

Figure . Try Catch -> Replace Text in Document

After the text replacement, I added a **Save Document as PDF activity** to save the edited word document as a PDF in the email attachments folder for the [EmailAutomation workflow](#_Email_Automation). I saved it to the specified workfiles folder path and specific file name with the current date and checked the Replace Existing property to replace the word document just in case there are more than 1 email to be sent on the same day.

Graphical user interface, text, application, email

Description automatically generated

Figure . Save Document as PDF

This will therefore conclude the [PDFAutomation workflow](#_PDF_Automation).

The resulting PDF will look as such:

Graphical user interface, text, application, email

Description automatically generated

Figure . Resulting PDF document

# Email Automation

Lastly, I have created a workflow to automate the sending of an email that will be sent to my reporting personnel. This part of the process fulfills one of the required solution techniques to be used in the assignment – **Email Automation**. To reiterate, the necessary information is summarized as stated:

* My role in the organization
* Interest being researched (Meta Platforms, UNSDG Goal 9)
* Explanation of which genre is selected
* Key statistical indicators that support the genre selection
* Complete information from the Ticker, Investment budget and Company Profile extracted from the PDF Automation process

First, I created a set of pre-defined variables:

* **Email** would be my email
* **savingFolderPath** would be the full path of the email attachments folder containing the PDF created from the [PDFAutomation workflow](#_PDF_Automation) and the image of the data visualization chart from the [DataAnalysis workflow](#_Data_Analysis_&)



Figure . Pre-define variables

I used the assign activity to first assign the files in the **EmailAttachments** folder to a variable, **mailAttachments**.

Graphical user interface, text, application

Description automatically generated

Figure . Get files in email attachments folder and assign to mailAttachments

Then, I will use a **Send Outlook Mail Message activity** to send the email to me and CC to my teammates. The subject is as is, and the body will contain the information as stated before:

* "Role: Chief Operating Officer (COO)"
* "Interest Researched: SDG Goal 9 - Industry, Innovation and Infrastructure"
* "I believe that companies should focus on building resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation."
* "Key statistical indicators: P/E, P/B, ROI, ROE, ROA, NPM, SDG Development Statement and ESG Risk Rating"
* "Attached contains Column Chart Screenshot and Investment Portfolio."

Graphical user interface, text, application, email

Description automatically generated

Figure . Send Outlook Mail Message activity

When we look at the activity’s properties, we have to specify the **AttachmentsCollection** which is the **mailAttachments** variable, input the account which is the pre-defined variable **Email** and set the receiver to me and my teammates emails are to be placed for cc.

Graphical user interface, application

Description automatically generated

Figure . Send Outlook mail message properties

Then, log the completion of the [EmailAutomation Workflow](#_Email_Automation) has been completed.

Upon running the solution, the email would look like this for me and my teammates’ Outlook inboxes:

Graphical user interface, text

Description automatically generated

Figure . Resulting email

After running the solution, the user can review the logs to see what has went wrong in the process. In this case, the SDG Goal 9 Progress Statement was too long to replace the text and the warning has been issued. The rest of the process seemed to go smoothly.

Text

Description automatically generated with medium confidence

Figure . Output logs

This would therefore conclude the assignment.

# Reflections

I feel quite satisfied for what I have done in this assignment, but there are a lot of things I could have done to improve it. One of them would be to implement UiPath Orchestrator to schedule and start the process.

I do feel that I have learnt a lot for the basics of robotic process automation with UiPath. But I do think that I need more practice for the different activities as well as how I create the workflows and tackle the problem at hand. Specifically, I could have learnt debugging and error handling better, and especially for PDF and Word automation.

Aside from the UiPath theory and practical that I have learnt in class, I had to do more self-learning at home to further familiarize myself with the techniques to carry out various problems in automation like learning the various C# functions that will help carry out functions that are not found in the provided Visual Basic (VB) activities in UiPath. Moreover for this assignment, I had to familiarize myself with the financial ratios and the sustainability development goals in order to know how to compare the companies with my teammates and choose the best 2 as required in the assignment for the group component.

I do feel that I have managed to stay abreast of the technological developments (robots and automation) and have also managed to touch and learn about financial terms and the like.

# UiPath Certification

In order to complete this assignment, I had to learn the necessary techniques to create a Robotic Process Automation solution from **UiPath’s RPA Developer Foundation** course ([see References to access the link](#_References)). This satisfies one of the requirements to show the proof of completion certificate document of the RPA Developer Foundation course.

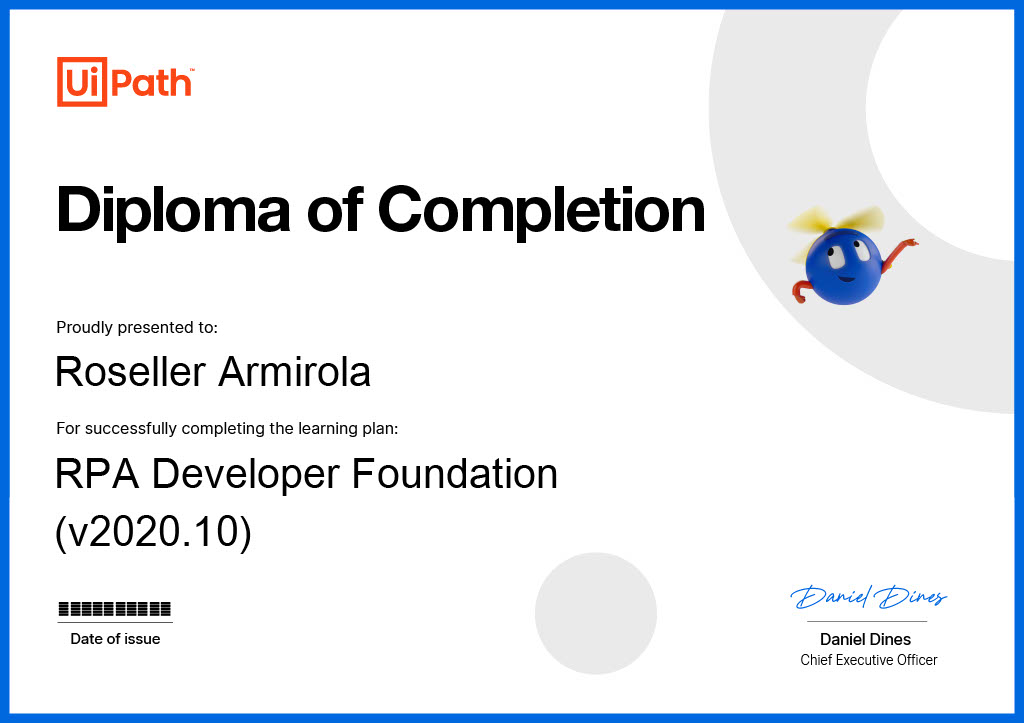


Figure . RPA Developer Foundation course diploma of completion

# References

* Financial Modeling Prep: <https://site.financialmodelingprep.com/>
* CSI Market: <https://csimarket.com/>
* Meta Investor Relations Website: <https://investor.fb.com/esg-resources/default.aspx/>
* Sustainalytics – Meta Platforms: <https://www.sustainalytics.com/esg-rating/meta-platforms-inc/1028643709>
* RPA Developer Foundation course by UiPath: <https://academy.uipath.com/learning-plans/rpa-developer-foundation>

# Appendices

Appendix A – Software(s) Used

* UiPath Studio
* Adobe Acrobat
* Microsoft Word
* Microsoft Excel
* Microsoft Outlook
* Chrome

Appendix B – Additional UiPath Package(s) Used

* BalaReva.EasyDataTable.Activities
* BalaReva.DataTable.Activities
* BalaReva.Excel.Activities
* UiPath.Word.Activities
* UiPath.Mail.Activities
* UiPath.PDF.Activities