INTELLIGENT AUTOMATION

CORPORATE INTERNSHIP @ JP MORGAN CHASE & CO.

VECTOR BUILDING – BENGALURU

DURATION – 4TH JULY TO 16TH DECEMBER

Manager: Ms. Padmashree Sandeep
(Corporate Sector : Intelligent Automation)

Mentor: Prof. Harlal Singh Mali
(Dept. of Mechanical Engineering)

Presented by: Shailesh Suthar
2019UME1168

CONTENTS:

J.P.Morgan

PROJECT APPROACH - GANTT CHART:

Sr. No.	Activity	Key Deliverables	Timeline (22 weeks)																					
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	Background understanding of JPMC	Welcome session																						
		introductory seesion and interaction with industry expert																						
		All building exploaration of JPMC																						
2	Project 1: Automation of Peru and Colombia daily risk report	Apply for all the access || basic brief about the project from stakholders																						
		Development of Peru daily risk report			Development							UAT				Prod Parallel			Deployment					
		Development of Colombia daily risk report									Development						UAT		Prod Parallel			Deployment		
3	Project 2: Development of Utility to automatically arrange the outlook meeting invite	Apply for all the access of sharedrive || basic brief about the project from stakholders																						
		Development of Automation														Development				UAT		Pr.	Depl.	

J.P.Morgan

EQUIVALENCE:

sr no	Code	Subject	Credit	Equivalance							
				0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
1	CET 432	Numerical Methods	3	1			2	3	2 & 4		
2	HST 408	Soft Skills & Personality Development	3	1			2	3	3		
3	MET 411	Finite Element Methods	4	1			2		3		1
4	MET 426	Mechatronic Design	4	1			2		3		1

Peru Daily Risk Report	Development	1
	UAT	2
	Prod Parallel	3
	Deployment	4

PyautoPDF	Development	1
	UAT	2
	Deployment to PiP Repo	3

Colombia Daily Risk Report	Development	1
	UAT	2
	Prod Parallel	3
	Deployment	4

Outlook meeting scheduling platform	Development	1
	UAT	2
	Prod Parallel	3
	Deployment	4

Python Libraries Documentation	Openpyxl	1
	win32com	2
	excel2img	3
	xlsxwriter	4

Alteryx	Foundation Certification	1

J.P.Morgan

BACKGROUND UNDERSTANDING OF JPMC:

Jamie Dimon – Chairman & CEO

Consumer Business	Wholesale Business		
Consumer & Community Bank (CCB)	Commercial Bank (CB)	Corporate & Investment Bank (CIB)	Asset & Wealth Management (AWM)
• Consumer Banking • Business Banking • Mortgage Banking • Commerce Solutions • Auto Finance	• Middle Market Banking • Corporate Client Banking • International Banking • Real Estate Banking • Community Development Banking	• Corporate Banking • Investment Banking • Treasury Services • Markets & Investor Services	• Investment (Asset) Management • Wealth Management

Corporate Sector

……

……..

Intelligent Automation

J.P.Morgan

PROJECT - 1

AUTOMATE THE DAILY RISK REPORT FOR PERU AND COLOMBIA

OBJECTIVE:

- At JPMC, the business teams in the global risk and compliance position exerted considerable work and time to prepare the daily report for the examination of the market, capital, and several other elements.
- In light of these considerations, the objective of this project was to develop a one-of-a-kind solution for all teams to generate their manual reports with a single click, so that key members of staff can devote more time to value-added activities that improve business insights and drive better decisions.

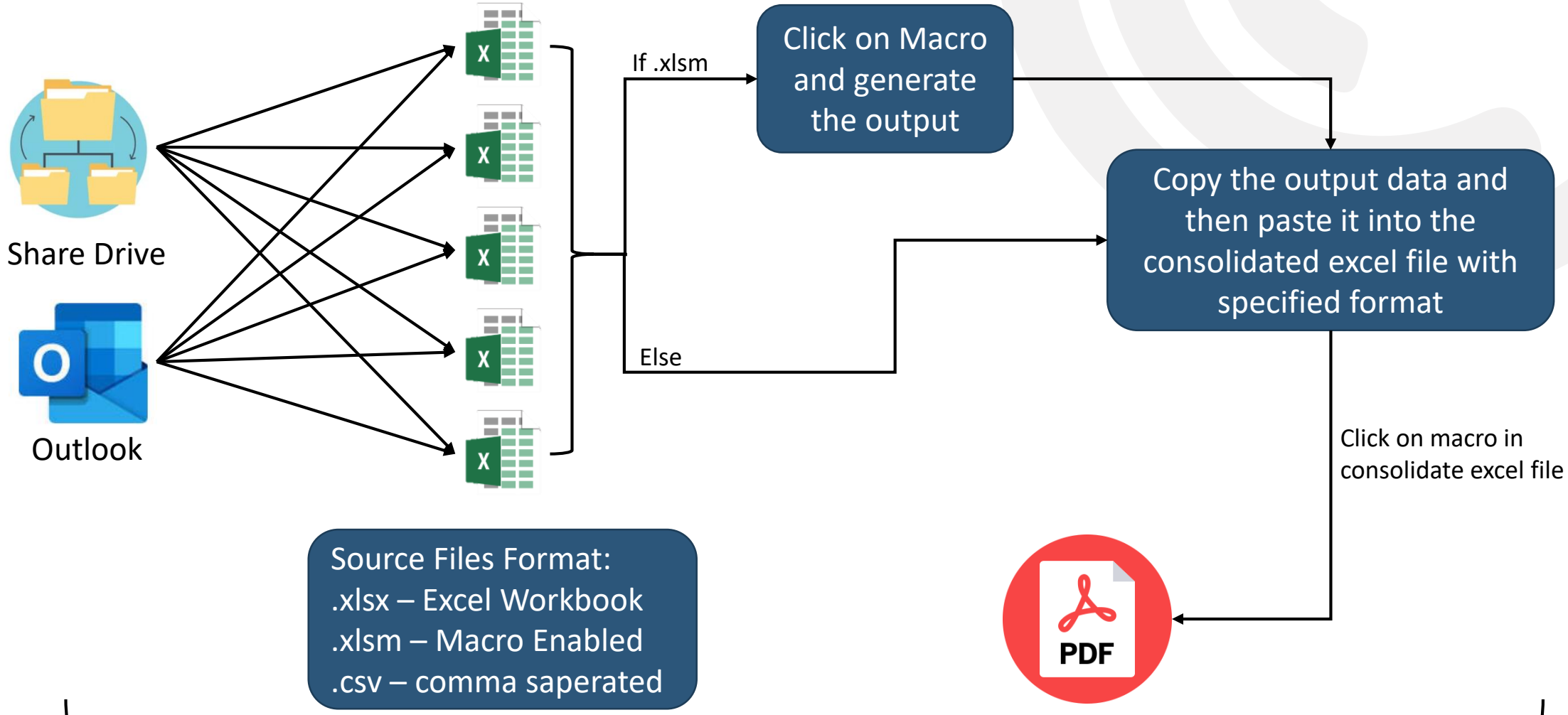
J.P.Morgan

OVERVIEW:

- There were two UT, one for Peru and one for Colombia.
- Both of them had a daily risk report.
- Each report was made by hand, and it took about five to six hours per day to do so.
- There were a lot of formatting issues with this report, such as text and cell colour coding, bold text, italic text, cell merging, and calculations that used two or more Excel files.
- The goal of this whole automation was to develop a unique Python programme that anyone could use to automate their business report in less time.
- Considering that I developed a Python interface for Peru and Colombia report and a unique library name called **“PyautoPDF”** for all around the global team.

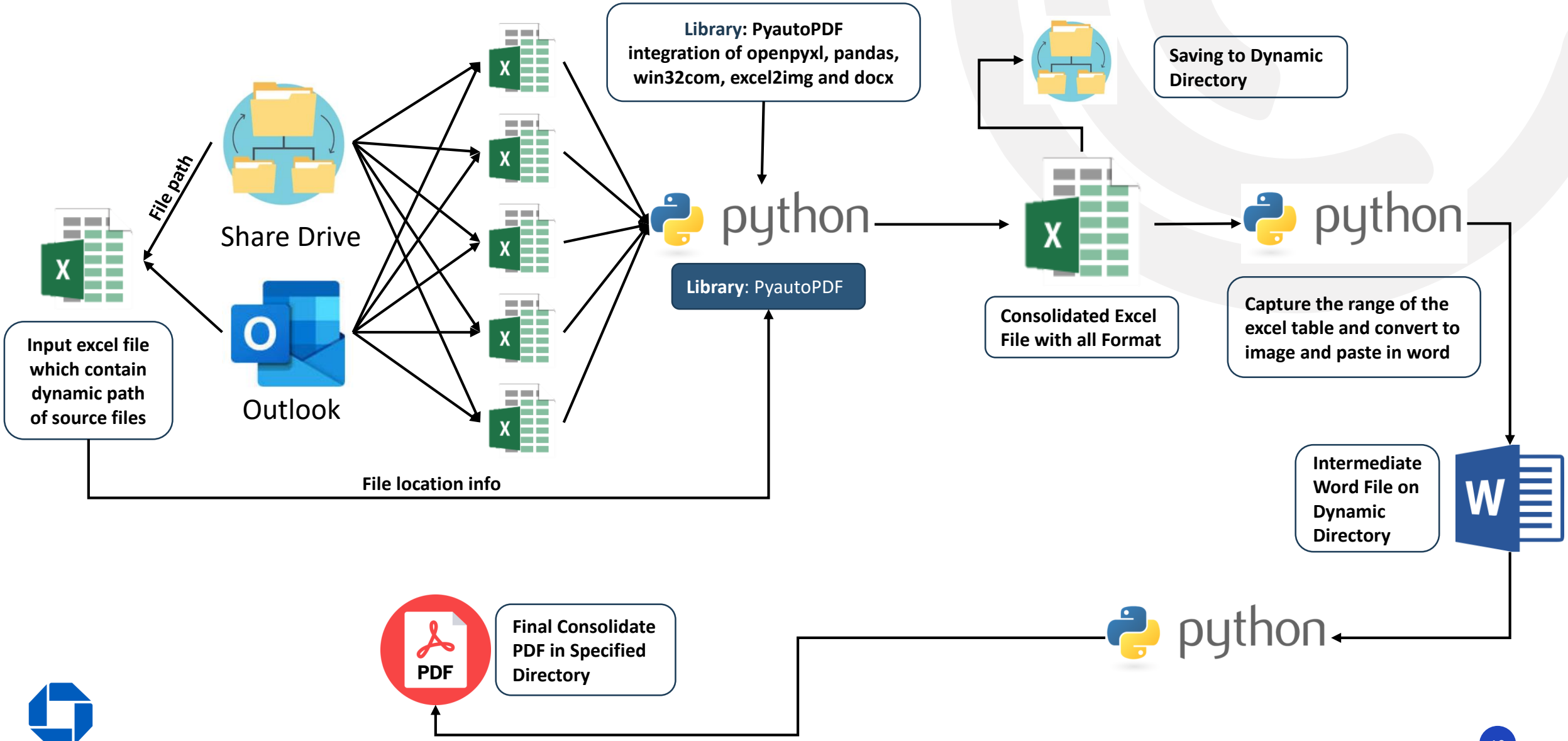
J.P.Morgan

MANUAL PROCESS OF REPORT DEVELOPMENT:



Expected Time Consumption: 5hr/day

AUTOMATION PROCESS OF REPORT DEVELOPMENT:



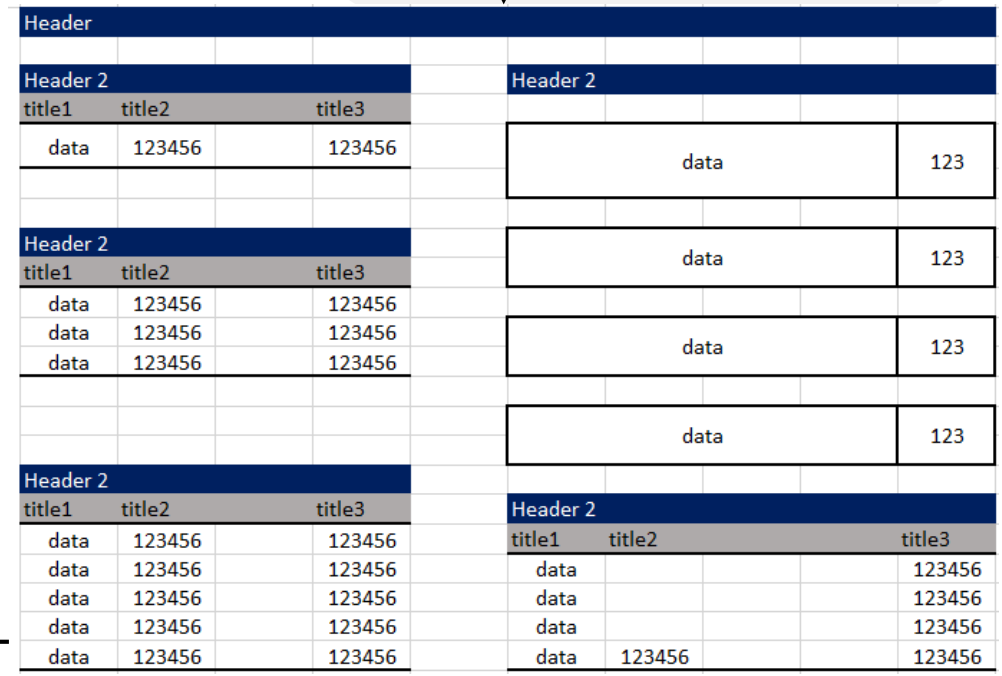
J.P.Morgan

TOOLS & TECHNOLOGY:

- **Software for monitoring the daily work:**
  - Atlassian Jira Board
- **Tools & Library:**
  - Python programming language
  - PyautoPDF
- **PyautoPDF:**
  - Integration of openpyxl, win32com, excel2img, docx, pandas

python

Library: PyautoPDF, Win32com, excel2img



PDF

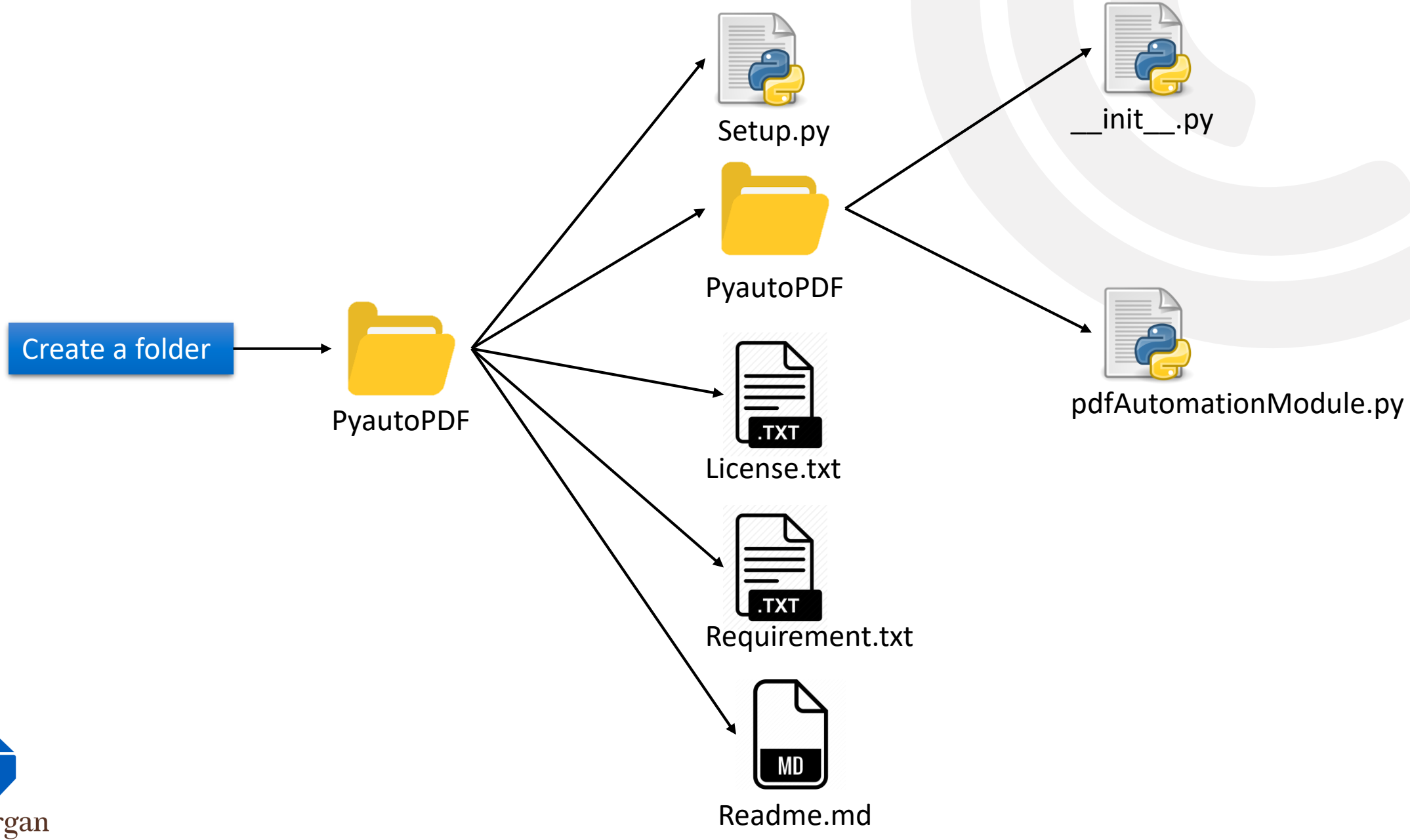
J.P.Morgan

PYTHON LIBRARY DEVELOPMENT || PYAUTOPDF:

- The primary goal of this library creation was to reduce non-value-added operations, code complexity, provide a better user experience, and optimise time consumption during report development using base libraries.
- PyautoPDF was the integration of python libraries name called openpyxl, win32com, excel2img, docx and pandas.

J.P.Morgan

PYAUTOPDF DEVELOPMENT:



J.P.Morgan

```python
from setuptools import setup, find_packages
import codecs
import os

here = os.path.abspath(os.path.dirname(__file__))

with codecs.open(os.path.join(here, "README.rst")) as fh:
    long_description = "\n" + fh.read()

VERSION = '0.1.5'
DESCRIPTION = 'Business Report Automation'

# Setting up
setup(
    name="PyautoPDF",
    version=VERSION,
    author="Shailesh Suthar",
    author_email="shaileshsuthar676@gmail.com",
    description=DESCRIPTION,
    long_description_content_type="text/markdown",
    license="MIT",
```

Setup.py

```markdown
Introduction
------------

PyautoPDF is a Python library to read/write Excel files which includes the

Mailing List
------------

Mail at - shaileshsuthar676@gmail.com

Sample Code::

    # import the library
    from PyautoPDF.PdfAutomationModule import PdfAutomation
    import PyautoPDF

    # create excel workbook
    wb = PyautoPDF.create_wb()

    # create a sheet at index zero in the existing workbook
    sheet1 = PyautoPDF.create_sheet(wb=wb, sheet_name='Sheet1', index=0)

    # variable declaration for sheet
    sheet1 = PdfAutomation(target_sheet=sheet1)
```

Readme.md

```python
class PdfAutomation:
    def __init__(self, target_sheet, heading1_color='749CBA', heading2_color='E0E3E4', text_size_h=12,
                 comment_color='DBECF0'):
        self.target_sheet = target_sheet
        self.heading1_color = heading1_color
        self.heading2_color = heading2_color
        self.text_size_h = text_size_h
        self.font = Font(name='Arial', size=self.text_size_h, bold=True, italic=False, vertAlign=None,
                         color='00FFFFFF')
        self.thin = Side(border_style='thin', color='000000')
        self.vertical_text = 'Side Vertical Text'
        self.vertical_text_cover_page = 'Cover Page Side Vertical Text'
        self.comment_color = comment_color
        """
        :param target_sheet: this is a variable that refers to the excel file on which we want to perform operatio
        :param heading1_color: this is a variable of color code for primary heading of the table
        :param heading2_color: this is a variable of color code for secondary heading of the table
        """

    def adjust_columns(self, col_width):
        for i in col_width:
            self.target_sheet.column_dimensions[i].width = col_width[i]
        """
        With the help of this function we can adjust the columns width of a individual excel sheet.
        """
```

pdfAutomationModule.py

```python
import openpyxl
import pandas as pd
from openpyxl.styles import PatternFill, Border, Side, Alignment, Font
from openpyxl.formatting.rule import CellIsRule


def create_wb():
    wb = openpyxl.Workbook()
    return wb


def create_sheet(wb, sheet_name, index):
    wb.create_sheet(title=sheet_name, index=index)
    target_sheet = wb[sheet_name]
    return target_sheet


def just_open(filename):
    import win32com.client
    xl_app = win32com.client.Dispatch('Excel.Application')
    xl_app.Visible = False
    xl_book = xl_app.Workbooks.Open(filename)
    xl_book.Save()
    xl_book.Close()
```

pdfAutomationModule.py

```
MIT License

Copyright (c) 2023 PyautoPDF
author: Shailesh Suthar
Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

License.txt

```python
from PyautoPDF.PdfAutomationModule import create_wb, create_sheet
from PyautoPDF.PdfAutomationModule import PdfAutomation
```

__init__.py

```
openpyxl~=3.0.10
setuptools~=65.6.3
pandas~=1.5.2
```

Requirement.txt

J.P.Morgan

LIBRARY INTEGRATION TO THE PIP REPOSITORY:

- pip is a package management tool created primarily for installing Python packages from the online Python Package Index (commonly known as PyPI).
- It is the method used the most frequently to install Python packages.
- The integration of python library into the open source pip repository was to improve the user accessibility and easy to work on it.
- For integrating the library into pip repository required preinstalled library as well as few commands to do so.

LIBRARY INTEGRATION TO THE PIP REPOSITORY:

- Referencing library installation:

```
(venv) PS C:\Users\shailesh suthar\PycharmProjects\PyautoPDF> pip install twine
Requirement already satisfied: twine in c:\users\shailesh suthar\pycharmprojects\pyautopdf\venv\lib\site-packages (4.0.2)
Requirement already satisfied: requests-toolbelt!=0.9.0,>=0.8.0 in c:\users\shailesh suthar\pycharmprojects\pyautopdf\venv\lib\site-packages (from twine) (0.10.1)
Requirement already satisfied: readme-renderer>=35.0 in c:\users\shailesh suthar\pycharmprojects\pyautopdf\venv\lib\site-packages (from twine) (37.3)
Requirement already satisfied: rfc3986>=1.4.0 in c:\users\shailesh suthar\pycharmprojects\pyautopdf\venv\lib\site-packages (from twine) (2.0.0)
Requirement already satisfied: importlib-metadata>=3.6 in c:\users\shailesh suthar\pycharmprojects\pyautopdf\venv\lib\site-packages (from twine) (5.2.0)
Requirement already satisfied: requests>=2.20 in c:\users\shailesh suthar\pycharmprojects\pyautopdf\venv\lib\site-packages (from twine) (2.28.1)
Requirement already satisfied: rich>=12.0.0 in c:\users\shailesh suthar\pycharmprojects\pyautopdf\venv\lib\site-packages (from twine) (12.6.0)
Requirement already satisfied: keyring>=15.1 in c:\users\shailesh suthar\pycharmprojects\pyautopdf\venv\lib\site-packages (from twine) (23.13.1)
```

- Commands execution in terminal of PyCharm.

```
(venv) PS C:\Users\shailesh suthar\PycharmProjects\PyautoPDF> python .\setup.py sdist bdist_wheel

(venv) PS C:\Users\shailesh suthar\PycharmProjects\PyautoPDF> twine upload dist/*
```

After the “twin upload dist/*” command enter your username and password of your pip repository, if you don’t have, go to pip repository and create a new account.

J.P.Morgan

LIBRARY INSTALLATION IN ANY LOCAL SYSTEM:

The Python Package Index (https://pypi.org/) hosts many useful third-party modules, and Pip is a package manager that makes it easy to install them.

- Launch a Command Prompt window on Windows and type:

```
pip install --user PyautoPDF
```

- If you're using macOS or Linux, you need to open a Terminal window and use pip3 instead of pip, as pip is for the older Python 2 version. Ubuntu Linux's default Python distribution does not include the pip package installer. To set it up, use a Terminal session and type “sudo apt-get instal python3-pip”.

```
python -m pip install –user PyautoPDF
```

- On macOS and Linux, run:

```
python3 -m pip install –user PyautoPDF
```

J.P.Morgan

SAMPLE CODE :

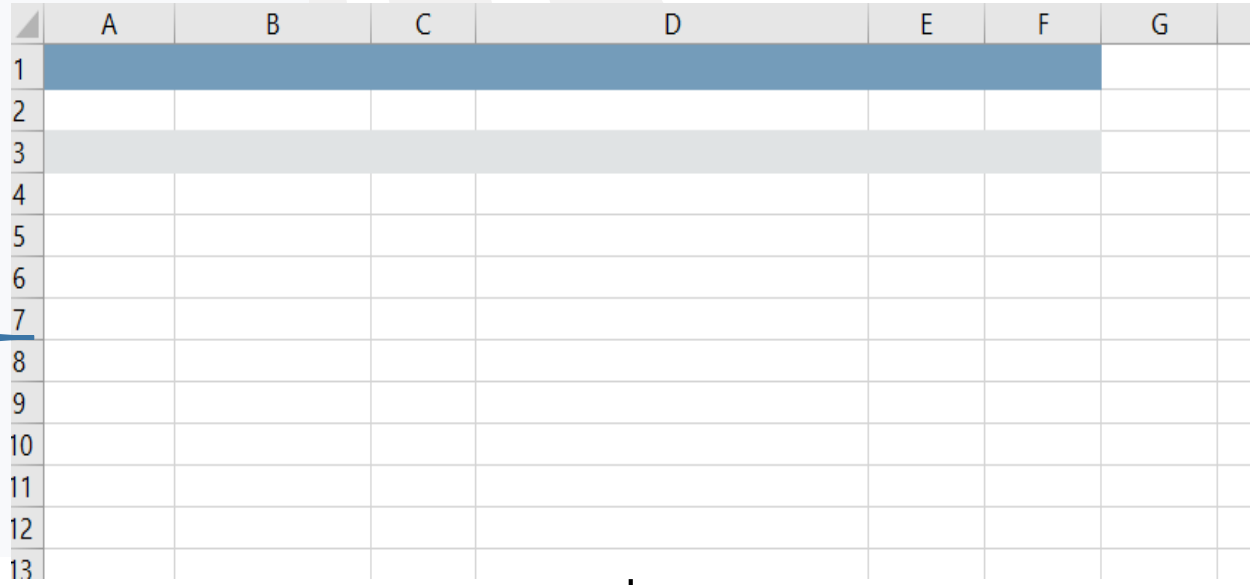
The Python Standard Library is a set of **pre-written script modules that may be imported into a Python application to speed up and streamline routine tasks.** You can utilise them in your script by "calling" or "importing" them at the very beginning of your code. Le see a sample code written below of how to use the PyautoPDF library.

- Open any IDE in your local preferably PyCharm, create a new project and one sample.py file it that folder.
- After that open PyCharm terminal and type the command for installation of PyautoPDF as per your operating system (refer to section 2.4.3 for installation).
- After successful installation of PyautoPDF from open source PiP repository, write the sample code inside the sample.py file as mentioned below.

SAMPLE CODE :

```python
# import the library
from PyautoPDF.PdfAutomationModule import PdfAutomation
import PyautoPDF

# create excel workbook
wb = PyautoPDF.create_wb()

# create a sheet at index zero in the existing workbook
sheet1 = PyautoPDF.create_sheet(wb=wb, sheet_name='Sheet1', index=0)

# variable declaration for sheet
sheet1 = PdfAutomation(target_sheet=sheet1)
```

```python
# applying a methods to do the color filling task
sheet1.color_filling(1, 1, 6, 8, 2, 1)

# applying a methods to adjust the column width
col_width = {'A': 10, 'B': 15, 'C': 8, 'D': 30}
sheet1.adjust_columns(col_width=col_width)

# saving the workbook
wb.save('test.xlsx')
```



test.xlsx

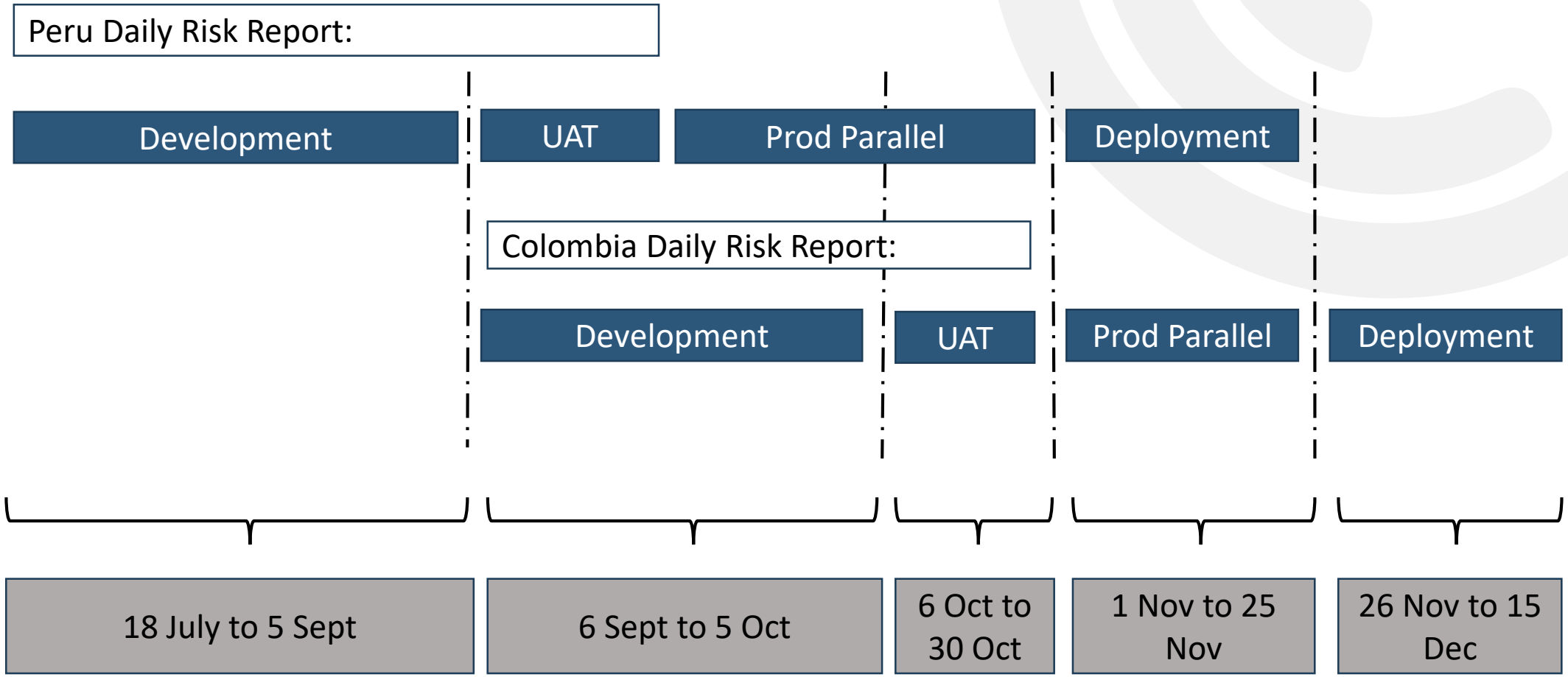
J.P.Morgan

BENEFITS:

- PiP repository is the great open source platform through which anyone can use it freely.
- Pip is the default package manager for Python.
- It enables the installation and administration of packages that are not a part of the Python standard library.
- The creation of PyautoPDF library and deploy it onto the PiP repository helps all the users as well as to the organisation to automate the existing report in less amount of time interval with minimal effort and less time consumption.
- The development of library was the futuristic unique solution towards the automation field.
- After the automation of both UT Peru as well as Colombia the time consumption for development had been reduced by 99.5 % redundant or non-value-added activities.

J.P.Morgan

PROJECT TIMELINE:



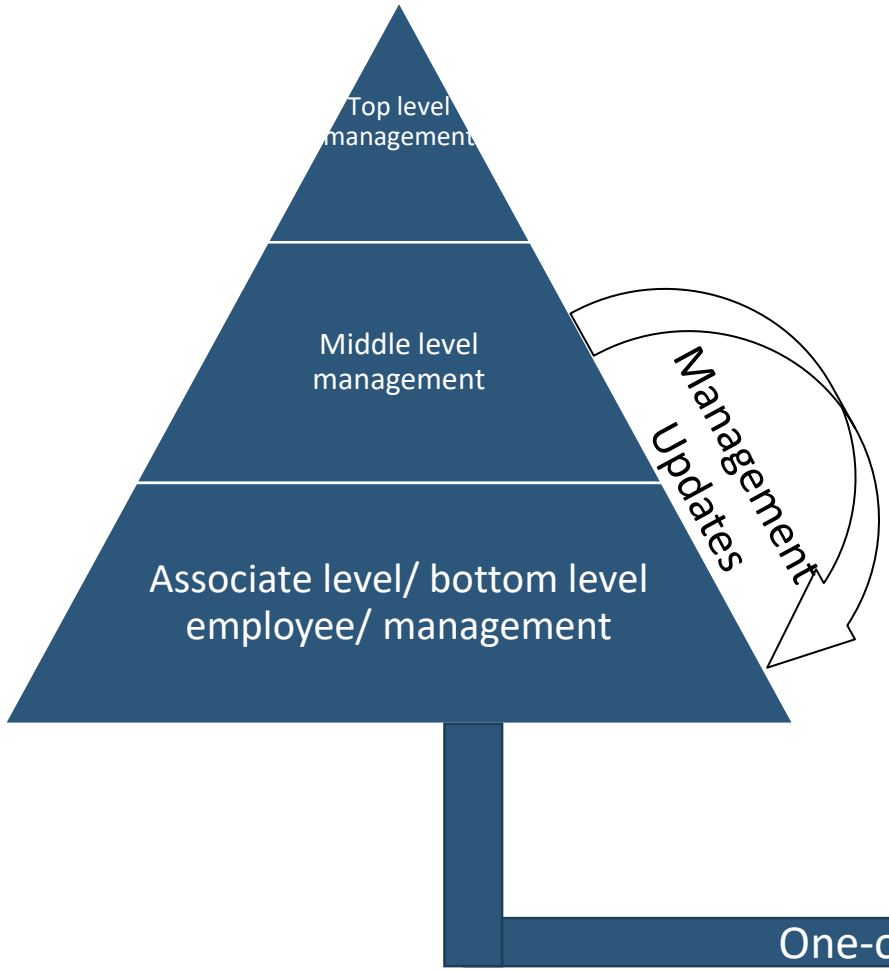
J.P.Morgan

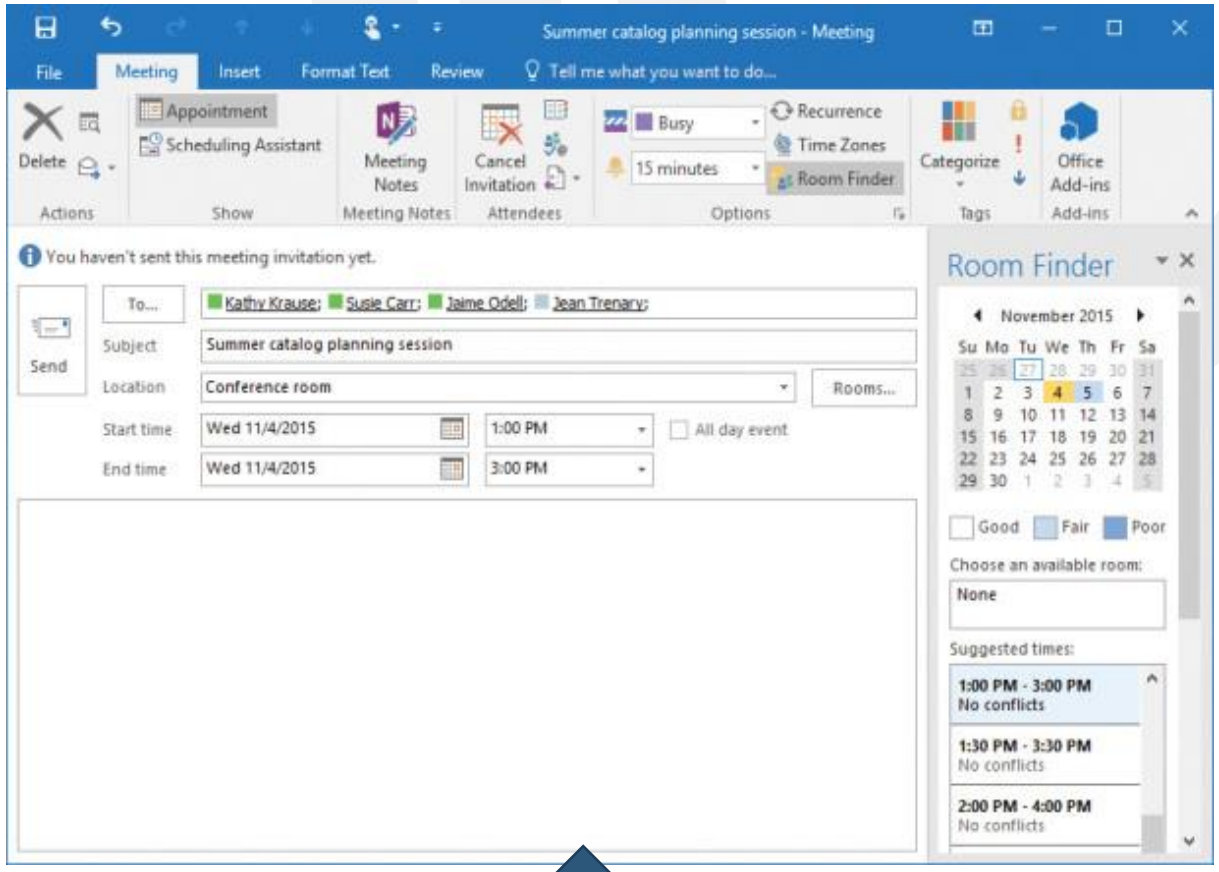
J.P.Morgan

PROJECT - 2

AUTOMATE THE BULK OUTLOOK MEETING SCHEDULING

BRIEF INTRODUCTION OF PROJECT

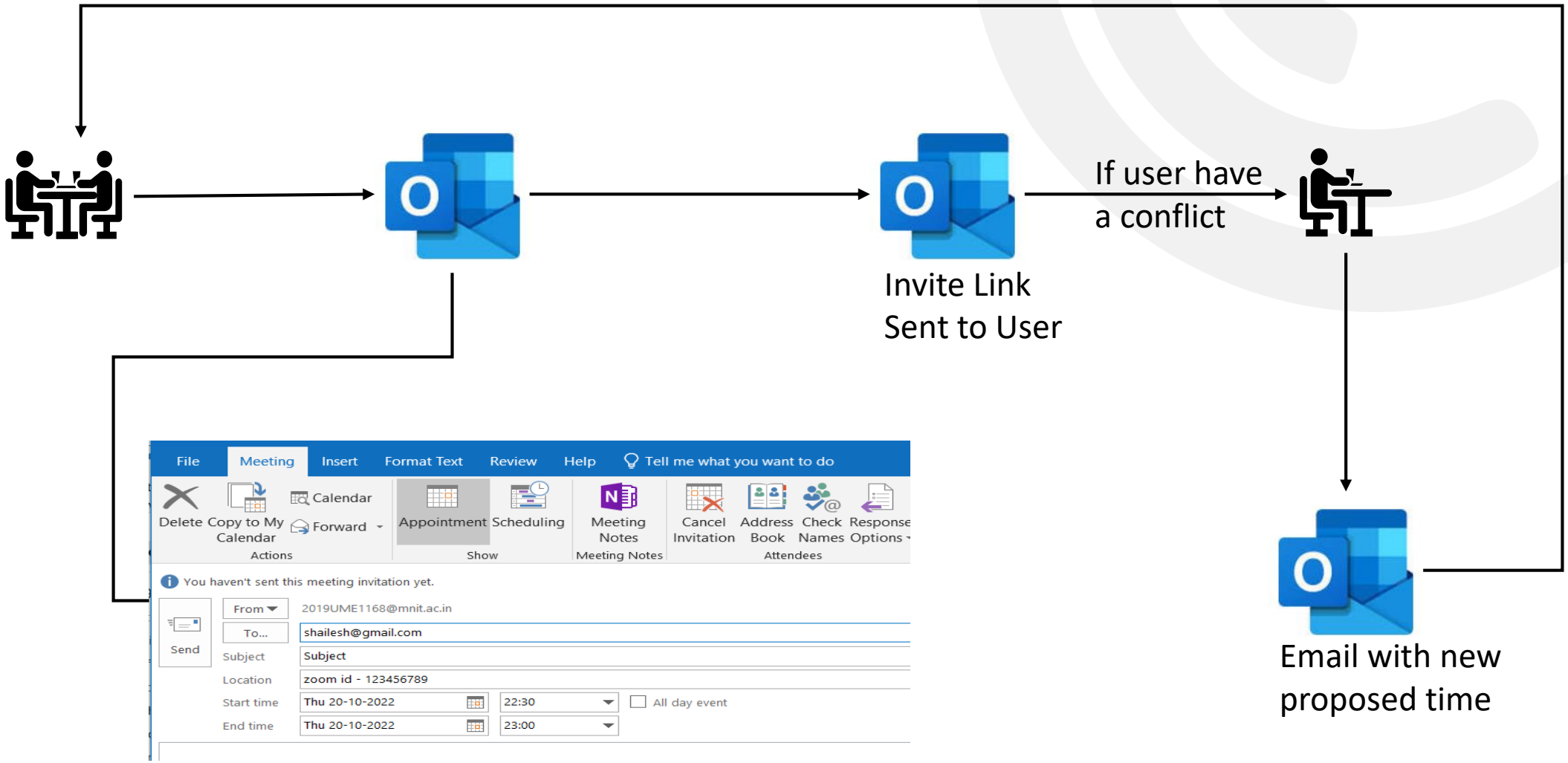




One-o-one meeting arrangement

J.P.Morgan

MANUAL PROCESS



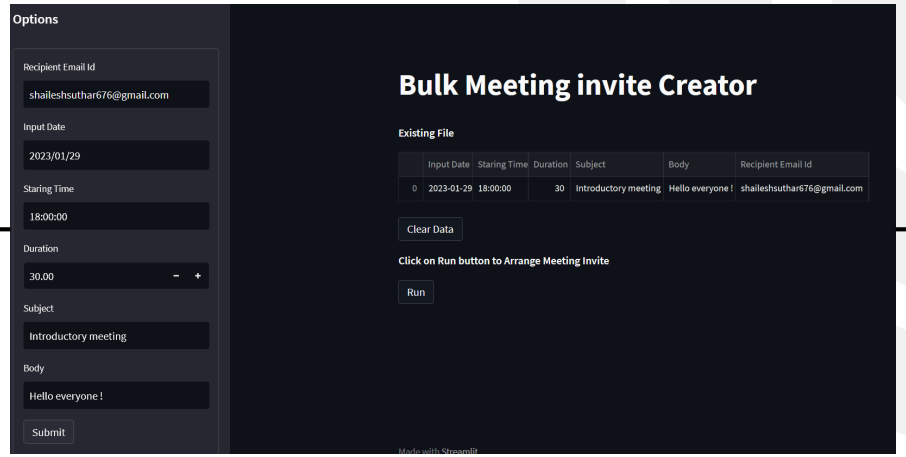
J.P.Morgan

AUTOMATION: ALGORITHM UNDERSTANDING

python

Using Streamlit

web page



manager

Hit run button by manager

Run

user

Team member will enter their free slot and hit Submit button, then the entry will reflect in the main table.

Meeting scheduled

One-o-one meeting has scheduled corresponding each entry

J.P.Morgan

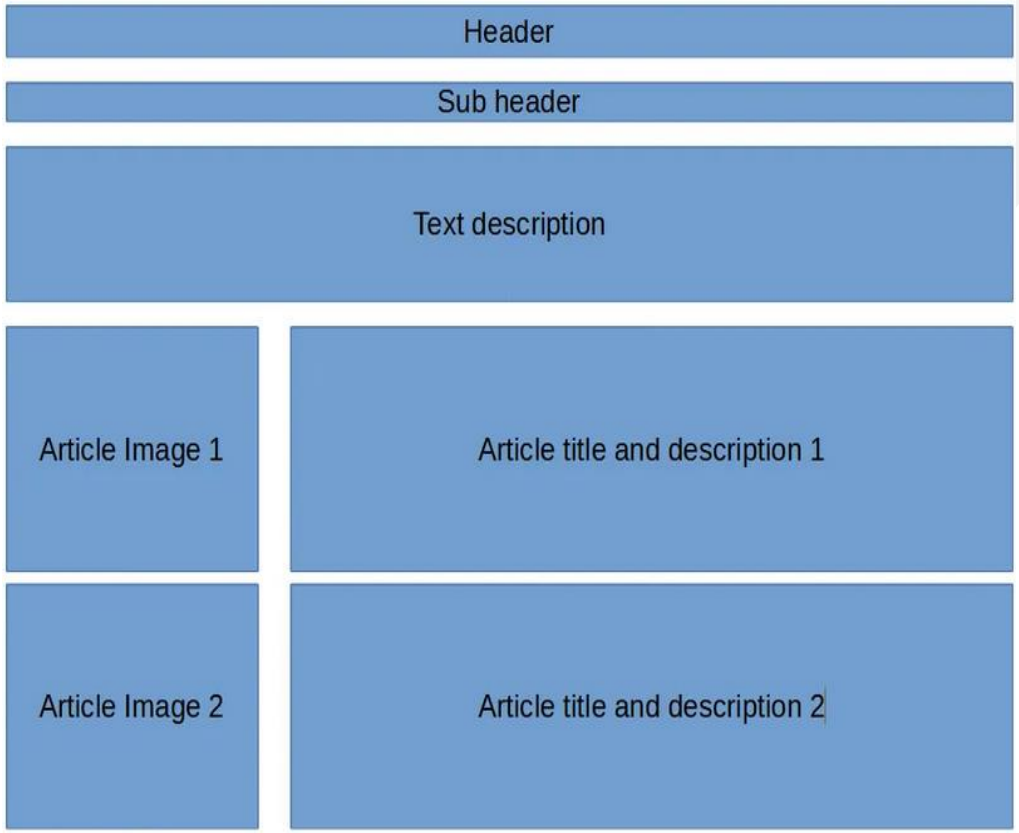
TOOLS & TECHNOLOGY:

- **Software for monitoring the daily work:**
  - Atlassian Jira Board
- **Tools & Library:**
  - Python programming language
  - Win32com
  - pandas
  - Streamlit

J.P.Morgan

DEVELOPMENT OF WEB PAGE USING STREAMLIT:

- Streamlit is an **open-source Python framework** that lets you make beautiful, interactive websites for Machine Learning and Data Science projects without having to know anything about web development. Streamlit is very easy to use.
- You need to know a very small amount of Python (almost nothing, to be honest), but you don't need to know HTML or JavaScript.
- Some very basic Markdown could also be helpful.



J.P.Morgan

DEVELOPMENT OF WEB PAGE:

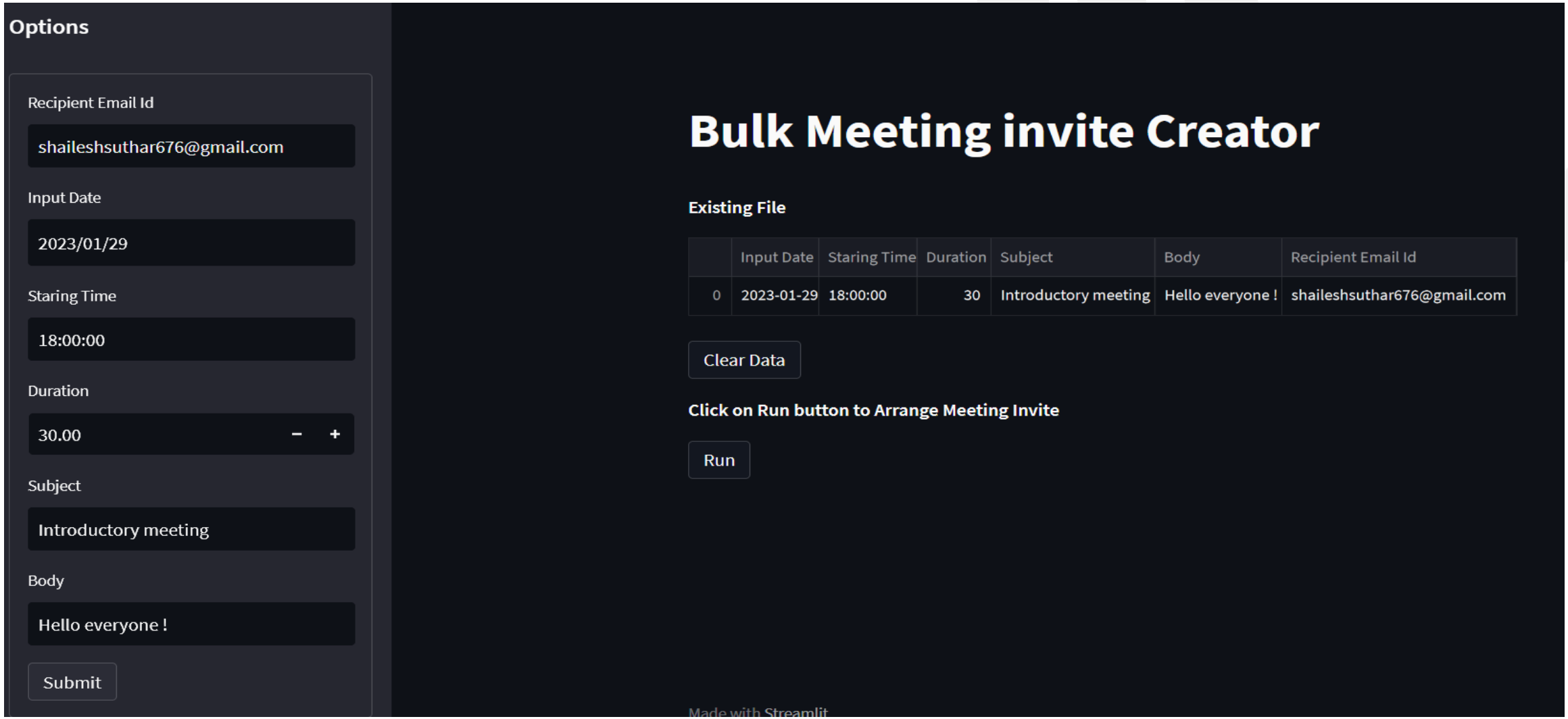
- Create a new project with the name "outlook_meeting_automation" in any IDE as similar to PyCharm.
- Go to File > Setting > Python Interpreter > click on “+” icon to install the required library.
- Search the library at search box like pandas, openpyxl, win32com, pythoncom, datetime, streamlit and hit install button.
- Create a “app.py” python file inside the main folder.

```python
import streamlit as st
import pandas as pd
import openpyxl
import datetime
import pythoncom

df = pd.read_excel('outlook_meeting_scheduling.xlsx')
df.sort_values(['Input Date', 'Staring Time'], inplace=True, axis=0, ascending=[True, True])
st.title('Bulk Meeting invite Creator')
st.markdown('**Existing File**')
st.write(df)

st.sidebar.header('Options')
options_form = st.sidebar.form('option_form')
email_id = options_form.text_input('Recipient Email Id')
date = options_form.date_input('Input Date', datetime.datetime.today())
s_time = options_form.text_input('Staring Time')
Duration = options_form.number_input('Duration')
subject = options_form.text_input('Subject')
body = options_form.text_input('Body')
add_data = options_form.form_submit_button()
clear_all_data = st.button('Clear Data')
st.markdown('**Click on Run button to Arrange Meeting Invite**')
set_meeting = st.button('Run')
```

```python
early_len = len(df["Input Date"])
# Delete duplicate rows based on specific columns
df = df.drop_duplicates(subset=["Input Date", 'Staring Time'], keep='first')
after_len = len(df["Input Date"])

if early_len > after_len:
    st.error('Please Select another Slot', icon="🚨")
df.to_excel('outlook_meeting_scheduling.xlsx', index=False)

def SendMeeting():
    import win32com.client as win32
    outlook = win32.Dispatch('Outlook.Application', pythoncom.CoInitialize())
    path = 'outlook_meeting_scheduling.xlsx'
    wb = openpyxl.load_workbook(filename=path)
    sheet = wb['Sheet1']
    total_rows = sheet.max_row
    for i in range(total_rows - 1):
        d = datetime.datetime.date(sheet.cell(row=i + 2, column=1).value)
        a = str(sheet.cell(row=i + 2, column=2).value).split(':')
        t = datetime.time(int(a[0]), int(a[1]), int(a[2]))
        fill_date = datetime.datetime.combine(d, t)
        delta_time = datetime.timedelta(hours=5, minutes=30)
        fill_subject = sheet.cell(row=i + 2, column=4).value
        duration = sheet.cell(row=i + 2, column=3).value
        location = '123456789'
        recipient = sheet.cell(row=i + 2, column=6).value
        fill_body = sheet.cell(row=i + 2, column=5).value
        outlook_sender = outlook.CreateItem(1)
        outlook_sender.Start = fill_date + delta_time
        outlook_sender.Subject = fill_subject
        outlook_sender.Duration = int(duration)
        outlook_sender.Location = location
        outlook_sender.Body = fill_body
        outlook_sender.MeetingStatus = 1
        outlook_sender.Recipients.Add(recipient)
        outlook_sender.Save()
        outlook_sender.Send()
```

J.P.Morgan

SAMPLE PLATFORM OF WORKING:



J.P.Morgan

BENEFITS:

- The benefits of the whole automation were to reduce the non-value-added activity belonging to the feedback looping in manual work and its even grows higher when the team grows, the reduction of time consumption with assisting the rescheduling work again and again based on conflicting with one schedule to another and after implementation of web platform there were not required any assistant to do the work.
  - The benefit of the web page was it works on different- different time zone.
  - This platform able to arrange a meeting in group or in person.
  - This web page able to remove the duplicate data entry with alert massage to the user.

J.P.Morgan

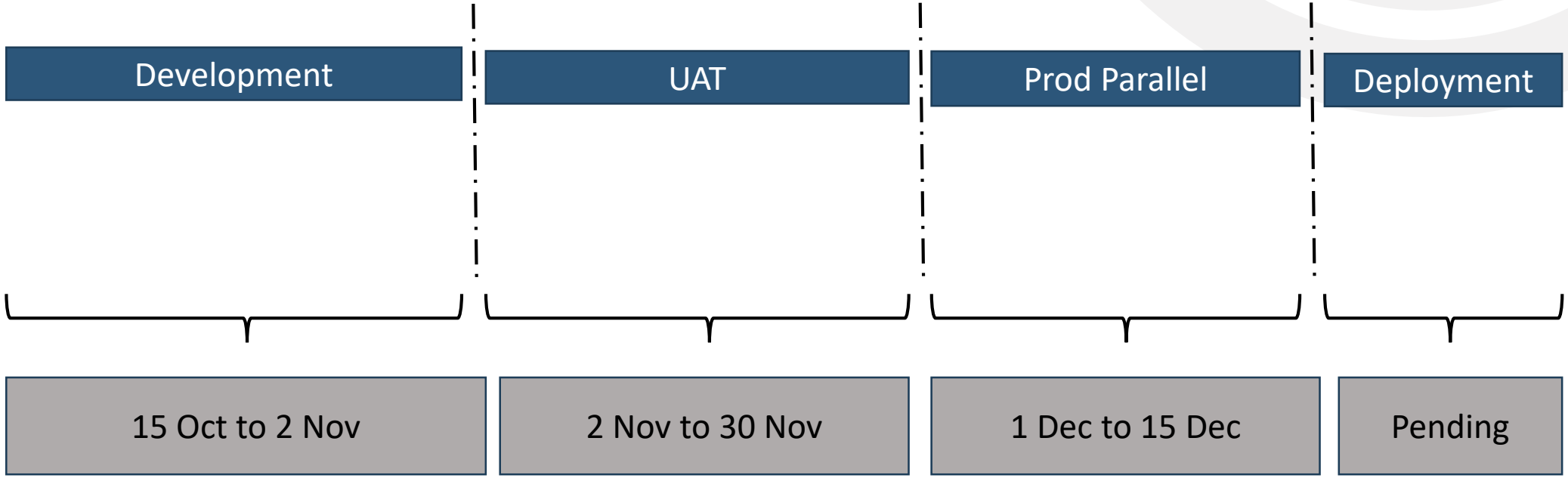
LIMITATION OF AUTOMATED WORK:

- There were few major critical factors which needs to be focus on that:
  - This web page didn’t have a user specific sign-in facility.
  - This web page was not useful when we wanted to arrange two or even more management meetings together at the same time slot, its only made for one management person at a time. The improvement work was running.

J.P.Morgan

PROJECT TIMELINE:

outlook meeting scheduling:



J.P.Morgan

CONCLUSION:

- The internship started off with an orientation that taught me about the company and its inner workings, the various departments and lines of business that make up the company.
- The interface creation and deployment, together with the elimination of all redundant components in the system, could not have been achieved without the pre-requisites in place.
- In "**Peru and Colombia Daily Report Development**“, I used programming skills to come up with a sturdy solution while still optimising the system.
- The goal of the internship was to automate an already-existing manual process using powerful technology so that other people might utilise it to save time throughout the day on similar processes.
- Along with the report automation I also worked in the field of web-development using a “streamlit” as saw how “**Outlook Meeting Scheduling**” can be done easily without any conflict and assistance of any person to reduce the tie wastage on non-value-added activities.

J.P.Morgan

REFERENCE:

- "Introduction to CreditMetrics - April 2, 1997 | Enhanced Reader."
- "Our Businesses | Job & Internships | JPMorgan Chase & Co." https://careers.jpmorgan.com/us/en/our-businesses (accessed Mar. 02, 2023).
- R. Mouly Potluri, Y. Batima, and K. Madiyar, "Corporate social responsibility: a study of Kazakhstan corporate sector," *Social Responsibility Journal*, vol. 6, no. 1, pp. 33–44, Mar. 2010, doi: 10.1108/17471111011024531/FULL/PDF.
- W. Hlawitschka and M. Tucker, "Wealth management: The relative importance of asset allocation and security selection," *Journal of Asset Management 2006 7:1*, vol. 7, no. 1, pp. 49–59, May 2006, doi: 10.1057/PALGRAVE.JAM.2240201.
- A. Bodnaruk, M. Massa, and A. Simonov, "Investment Banks as Insiders and the Market for Corporate Control," *Rev Financ Stud*, vol. 22, no. 12, pp. 4989–5026, Dec. 2009, doi: 10.1093/RFS/HHP043.
- R. Deyoung, W. C. Hunter, and G. F. Udell, "The past, present, and probable future for community banks," *Journal of Financial Services Research*, vol. 25, no. 2–3, pp. 85–133, 2004, doi: 10.1023/B:FINA.0000020656.65653.79/METRICS.
- A. M. Santomero, "Commercial Bank Risk Management: An Analysis of the Process," *Journal of Financial Services Research*, vol. 12, no. 2–3, pp. 83–115, 1997, doi: 10.1023/A:1007971801810/METRICS.
- P. B. Topalova, "Overview of the Indian Corporate Sector: 1989-2002." Apr. 01, 2004. Accessed: Mar. 02, 2023. [Online]. Available: https://papers.ssrn.com/abstract=878887

J.P.Morgan

REFERENCE:

- "Acceptance of agile methodologies: A critical review and conceptual framework," *Decis Support Syst*, vol. 46, no. 4, pp. 803–814, Mar. 2009, doi: 10.1016/J.DSS.2008.11.009.
- M. Golfarelli, S. Rizzi, and E. Turricchia, "Multi-sprint planning and smooth replanning: An optimization model," *Journal of Systems and Software*, vol. 86, no. 9, pp. 2357–2370, Sep. 2013, doi: 10.1016/J.JSS.2013.04.028.
- K. Schwaber, "What Is Scrum?", Accessed: Mar. 02, 2023. [Online]. Available: http://volaroint.com/posts/3.html
- S. Chaouch, A. Mejri, and S. A. Ghannouchi, "A framework for risk management in Scrum development process," *Procedia Comput Sci*, vol. 164, pp. 187–192, Jan. 2019, doi: 10.1016/J.PROCS.2019.12.171.
- W. : Www, M. Mahalakshmi, and D. M. Sundararajan, "International Journal of Emerging Technology and Advanced Engineering Traditional SDLC Vs Scrum Methodology-A Comparative Study," *Certified Journal*, vol. 9001, no. 6, 2008, Accessed: Mar. 02, 2023. [Online]. Available: www.ijetae.com
- N. Dean Meyer, "OFFICE AUTOMATION: A PROGRESS REPORT," *Office Technology and People*, vol. 1, no. 1, pp. 107–121, Jan. 1982, doi: 10.1108/EB022608.
- J. Hunt, "Working with Excel Files," pp. 249–255, 2019, doi: 10.1007/978-3-030-25943-3_21.
- V. Ragunathan, "Words of Agreement," *C++/CLI Primer*, pp. 3–4, 2016, doi: 10.1007/978-1-4842-2367-3_2.

J.P.Morgan

J.P.Morgan

2019UME1168@MNIT.AC.IN

HTTPS://GITHUB.COM/SHAILESHSUTHAR675