**Lab Exercise 7 - Reading Data from an Excel File Using openpyxl in RPA with Python**

**Objective**

This lab exercise focuses on learning how to read data from an Excel file using the openpyxl library, which is essential for automating repetitive tasks involving spreadsheets in Robotic Process Automation (RPA).

**Objective**

Learn to:

1. Open and read an Excel file.
2. Extract data from specific cells or ranges.
3. Process the data programmatically.

**Prerequisites**

1. Install the openpyxl library:

pip install openpyxl

1. Have an Excel file (example.xlsx) with sample data. Create it using the following data if it doesn't exist:

| **Product** | **Region** | **Sales** |
| --- | --- | --- |
| Laptop | North | 1200 |
| Tablet | South | 800 |
| Smartphone | East | 1500 |
| Laptop | West | 1000 |

Save it as **example.xlsx**.

**Step 1: Load and Access the Excel File**

from openpyxl import load\_workbook

# Load the workbook

workbook = load\_workbook("example.xlsx")

# Select the active sheet

sheet = workbook.active

# Print sheet title

print("Sheet Title:", sheet.title)

**Step 2: Read Specific Cells**

# Read data from specific cells

print("Cell A1:", sheet["A1"].value) # Header "Product"

print("Cell B2:", sheet["B2"].value) # "North"

print("Cell C3:", sheet["C3"].value) # Sales value 1500

**Step 3: Iterate Through Rows**

# Read all rows of data (excluding headers)

print("Data Rows:")

for row in sheet.iter\_rows(min\_row=2, max\_row=sheet.max\_row, values\_only=True):

print(row)

**Step 4: Extract Data for Processing**

# Extract data into a list of dictionaries

data = []

for row in sheet.iter\_rows(min\_row=2, values\_only=True):

product, region, sales = row

data.append({"Product": product, "Region": region, "Sales": sales})

print("\nExtracted Data:")

for record in data:

print(record)

**Step 5: Perform Basic Analysis**

# Calculate the total sales

total\_sales = sum(record["Sales"] for record in data)

print("\nTotal Sales:", total\_sales)

# Find the region with the highest sales

highest\_sales\_record = max(data, key=lambda x: x["Sales"])

print("\nRegion with Highest Sales:", highest\_sales\_record["Region"])

**Step 6: Handle Nonexistent Files Gracefully**

import os

# Check if the file exists

file\_path = "example.xlsx"

if os.path.exists(file\_path):

workbook = load\_workbook(file\_path)

print("File loaded successfully.")

else:

print(f"File '{file\_path}' does not exist.")

**Expected Output**

* Sheet title.
* Data read from specific cells.
* All rows of data printed line by line.
* Total sales calculated and displayed.
* Region with the highest sales identified.

**Key Concepts Covered**

1. Loading an Excel workbook using openpyxl.
2. Accessing specific cells and iterating through rows.
3. Extracting data for further processing.
4. Performing basic data analysis.
5. Implementing file existence checks for robustness.