1. Python program to Create a excel file

In [1]:

```python
import openpyxl
# Create a new workbook
workbook = openpyxl.Workbook()
# Select the active sheet
sheet = workbook.active
# Add data to cells
sheet['A1'] = 'Name'
sheet['B1'] = 'Age'
sheet['C1'] = 'City'
data = [
    ('ivin', 22, 'Thrissur'),
    ('Evana',22, 'Ernakulum'),
    ('Agnal', 23, 'Thrissur'),
    ('Lincy', 24, 'Ernakulum')
]
```

In [2]:

```python
for row in data:
    sheet.append(row)

# The workbook is saved
workbook.save('Data.xlsx')

print("Excel file is  created")
```

```
Excel file is  created
```

2. Python program for Import data from an excel file

In [3]:

```
import openpyxl
from tabulate import tabulate

# Load the Excel file
workbook = openpyxl.load_workbook('Data.xlsx')

# Select the active sheet
sheet = workbook.active

# Get the maximum row count
max_row = sheet.max_row

# Create an empty list to store the rows
table_data = []

# Iterate over the rows and columns
for row in range(1, max_row + 1):
    row_data = []
    for column in range(1, sheet.max_column + 1):
        cell_value = sheet.cell(row=row, column=column).value
        row_data.append(cell_value)
    table_data.append(row_data)

# Print the table
print(tabulate(table_data, headers="firstrow", tablefmt="grid"))
```

```
+--------+-------+-----------+
| Name   |   Age | City      |
+========+=======+===========+
| ivin   |    22 | Thrissur  |
+--------+-------+-----------+
| Evana  |    22 | Ernakulum |
+--------+-------+-----------+
| Agnal  |    23 | Thrissur  |
+--------+-------+-----------+
| Lincy  |    24 | Ernakulum |
+--------+-------+-----------+
```

3. Python program for Format data in excel sheet

In [4]:

```
from openpyxl.styles import Font, Alignment

# Load the Excel file
workbook = openpyxl.load_workbook('Data.xlsx')

# Select the active sheet
sheet = workbook.active

# Format the header row
header_row = sheet[1]
header_font = Font(bold=True)
for cell in header_row:
    cell.font = header_font
    cell.alignment = Alignment(horizontal='center')

# Format the data rows
data_font = Font(italic=True)
data_alignment = Alignment(horizontal='left')
for row in sheet.iter_rows(min_row=2):
    for cell in row:
        cell.font = data_font
        cell.alignment = data_alignment

# Save the modified workbook
workbook.save('formatted_Data.xlsx')
```

4. Python program for Prepare Yoshops Survey and Order excel charts Like = Pie Chart and Bar Chart Weekly, Monthly and Yearly Reports.

In [6]:

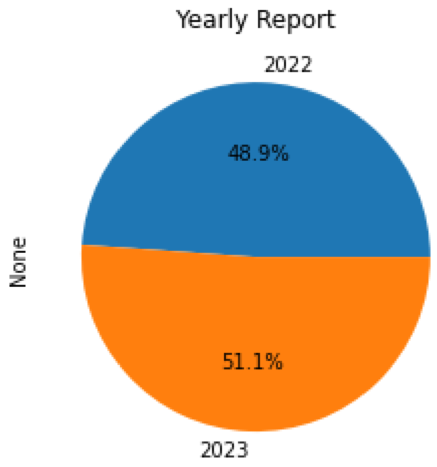
```
import pandas as pd

survey = pd.read_excel("Yoshops Survey.xlsx")
survey.head()
```

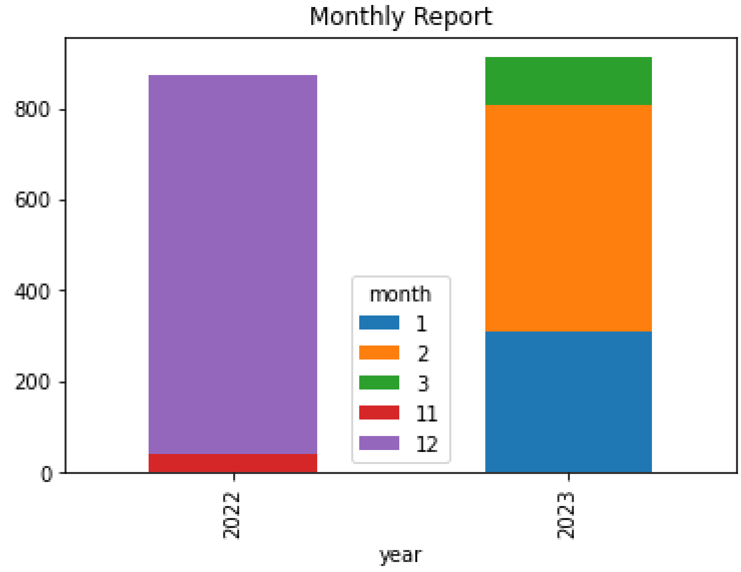
Out[6]:

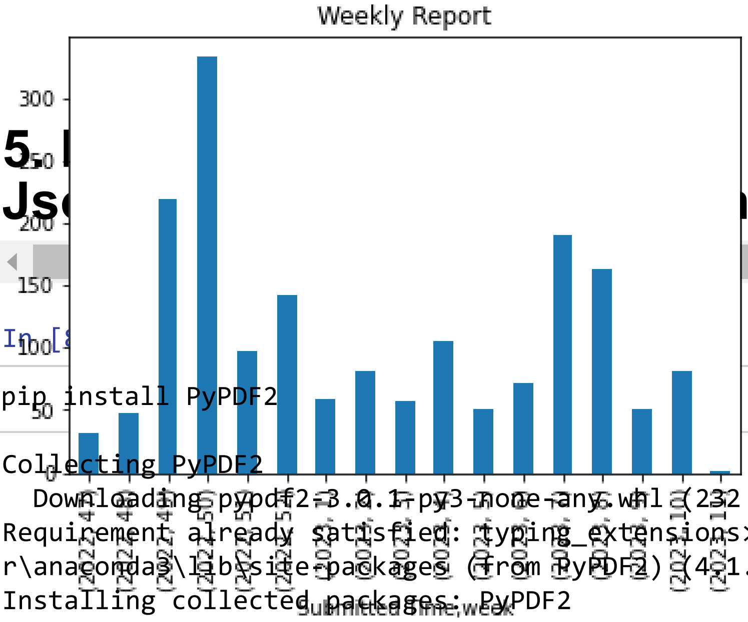
	S.NO	Submitted Time	1. Name	3. Location , City Name	4. What class do you study in ?	5. Which Price range for Online Tution for LKG to PG Monthly Fees You like must	6. Laptop and Mobile which Price range you like most	7. What is your Favourite food biryani essay?	8. What is Price Rang for 1kg Biryani(2 Person Eat) you like must	p
0	1	24-11-2022	Kavita Israni	No answer	No answer	undefined	No answer	Veg Biryani | undefined	149 | 199	
1	2	24-11-2022	Kunal Anand	No answer	No answer	undefined	No answer	Chicken Biryani	149	S
2	3	25-11-2022	Deepak parmal	No answer	Graduation	undefined	No answer	Chicken Biryani	99	Inv
3	4	25-11-2022	Nidhi Gupta	No answer	LKG to STD 5 | STD 6 to STD 10	undefined	No answer	Chicken Biryani	149	E
4	5	25-11-2022	Rohan Pandey	No answer	STD 6 to STD 10	undefined	No answer	Chicken Biryani	149	E

In [7]:

```
import matplotlib.pyplot as plt
# Convert the 'date' column to datetime format
survey['Submitted Time'] = pd.to_datetime(survey['Submitted Time'], format='%d-%m-%Y')

# Extract year, month, and week from the 'date' column
survey['year'] = survey['Submitted Time'].dt.year
survey['month'] = survey['Submitted Time'].dt.month
survey['week'] = survey['Submitted Time'].dt.isocalendar().week

# Yearly report
yearly_data = survey.groupby('year').size()
yearly_data.plot(kind='pie', autopct='%1.1f%%')
plt.title('Yearly Report')
plt.show()

# Monthly report
monthly_data = survey.groupby(['year', 'month']).size().unstack()
monthly_data.plot(kind='bar', stacked=True)
plt.title('Monthly Report')
plt.show()

# Weekly report
weekly_data = survey.groupby([survey['Submitted Time'].dt.year, survey['week']]).size()
weekly_data.plot(kind='bar', stacked=True)
plt.title('Weekly Report')
plt.show()
```







5. I mobile no from PDF, Js d save into MS excel

In [

```
pip install PyPDF2
```

```
Collecting PyPDF2
  Downloading pypdf2-3.0.1-py3-none-any.whl (232 kB)
Requirement already satisfied: typing_extensions>=3.10.0.0 in c:\users\use
r\anaconda3\lib\site-packages (from PyPDF2) (4.1.1)
Installing collected packages: PyPDF2
Successfully installed PyPDF2-3.0.1
Note: you may need to restart the kernel to use updated packages.
```

In [9]:

```
pip install python-docx
```

```
Collecting python-docx
  Downloading python-docx-0.8.11.tar.gz (5.6 MB)
Requirement already satisfied: lxml>=2.3.2 in c:\users\user\anaconda3\lib
\site-packages (from python-docx) (4.8.0)
Building wheels for collected packages: python-docx
  Building wheel for python-docx (setup.py): started
  Building wheel for python-docx (setup.py): finished with status 'done'
  Created wheel for python-docx: filename=python_docx-0.8.11-py3-none-any.
whl size=184507 sha256=ac725519af8f7e5366b74706b208367d093d6e7aa2792c8898a
0e29621378326
  Stored in directory: c:\users\user\appdata\local\pip\cache\wheels\83\8b
\7c\09ae60c42c7ba4ed2dddaf2b8b9186cb105255856d6ed3dba5
Successfully built python-docx
Installing collected packages: python-docx
Successfully installed python-docx-0.8.11
Note: you may need to restart the kernel to use updated packages.
```

In [13]:

```
import os
import re
import json
import xml.etree.ElementTree as ET
import PyPDF2
from docx import Document
import pandas as pd

def extract_mobile_numbers(text):
    pattern = r"\b(?:\+?(\d{1,3}))?[-. (]*(\d{3})[-. )]*(\d{3})[-. ]*(\d{4})\b"
    mobile_numbers = re.findall(pattern, text)
    return [''.join(number) for number in mobile_numbers]

def extract_mobile_numbers_from_pdf(file_path):
    try:
        with open(file_path, 'rb') as f:
            reader = PyPDF2.PdfReader(f)
            text = ""
            for page in reader.pages:
                text += page.extract_text()
        mobile_numbers = extract_mobile_numbers(text)
        return mobile_numbers
    except Exception as e:
        print(f"Error extracting mobile numbers from PDF: {e}")
        return []
def extract_mobile_numbers_from_json(file_path):
    try:
        with open(file_path, 'r') as f:
            data = json.load(f)
        text = json.dumps(data)
        mobile_numbers = extract_mobile_numbers(text)
        return mobile_numbers
    except Exception as e:
        print(f"Error extracting mobile numbers from JSON: {e}")
        return []

def extract_mobile_numbers_from_xml(file_path):
    try:
        tree = ET.parse(file_path)
        root = tree.getroot()
        text = ET.tostring(root, encoding='unicode')
        mobile_numbers = extract_mobile_numbers(text)
        return mobile_numbers
    except Exception as e:
        print(f"Error extracting mobile numbers from XML: {e}")
        return []

def extract_mobile_numbers_from_word(file_path):
    try:
        doc = Document(file_path)
        text = ' '.join([paragraph.text for paragraph in doc.paragraphs])
        mobile_numbers = extract_mobile_numbers(text)
        return mobile_numbers
    except Exception as e:
        print(f"Error extracting mobile numbers from Word: {e}")
        return []

def save_mobile_numbers_to_excel(mobile_numbers, output_file):
    df = pd.DataFrame({'Mobile Number': mobile_numbers})
    df.to_excel(output_file, index=False)
    print(f"Mobile numbers saved to {output_file}")

def main():
    directory_path = r"C:\Users\user\OneDrive\Desktop\internship\contact data.zip"
    mobile_numbers = []

    for root, dirs, files in os.walk(directory_path):
        for file_name in files:
            file_path = os.path.join(root, file_name)

            if file_name.endswith('.pdf'):
                mobile_numbers.extend(extract_mobile_numbers_from_pdf(file_path))
            elif file_name.endswith('.json'):
                mobile_numbers.extend(extract_mobile_numbers_from_json(file_path))
            elif file_name.endswith('.xml'):
                mobile_numbers.extend(extract_mobile_numbers_from_xml(file_path))
            elif file_name.endswith('.docx'):
                mobile_numbers.extend(extract_mobile_numbers_from_word(file_path))

    excel_file_path = 'mobile_numbers.xlsx'
    if mobile_numbers:
        save_mobile_numbers_to_excel(mobile_numbers, excel_file_path)
    else:
        print("No mobile numbers found in the files.")

if __name__ == "__main__":
    main()
```

```
No mobile numbers found in the files.
```

6.Prepare python programm for data cleaning process to removing unnecessary data

In [26]:

```
order = pd.read_excel("Yoshops_Order_List.xlsx")
order.head()
```

Out[26]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6
0	Name	Address 1	Adress 2	Adress 3	City	pincode	state
1	SURJEET SINGH KADIAN	695-35 JANTA COLONY	0	0	Rohtak	124001	Haryana
2	LAKSHMI NARAYANAN S	K M B L CEEBROS CENTRE 1ST FLR	39 MONTEITH ROAD EGMORE	0	Chennai	600008	Tamil Nadu
3	TARUN SINHA	C-1052 1ST FLOOR	SUSHANT LOK - I	0	Gurgaon	122001	Haryana
4	AVNISH SRIVASTAVA	Fltno207 208 Dhavalgiri	Neelkanth Vihar Kurla Terminus	Pipeline Rd Ghatkopar E	Mumbai	400077	Maharashtra

In [29]:

```
order=order.rename(columns={'Unnamed: 0':'Name','Unnamed: 1':'Address 1','Unnamed: 2':'A
```

In [30]:

```
order.head()
```

Out[30]:

	Name	Address 1	Adress 2	Adress 3	City	Pincode	State
0	Name	Address 1	Adress 2	Adress 3	City	pincode	state
1	SURJEET SINGH KADIAN	695-35 JANTA COLONY	0	0	Rohtak	124001	Haryana
2	LAKSHMI NARAYANAN S	K M B L CEEBROS CENTRE 1ST FLR	39 MONTEITH ROAD EGMORE	0	Chennai	600008	Tamil Nadu
3	TARUN SINHA	C-1052 1ST FLOOR	SUSHANT LOK - I	0	Gurgaon	122001	Haryana
4	AVNISH SRIVASTAVA	Fltno207 208 Dhavalgiri	Neelkanth Vihar Kurla Terminus	Pipeline Rd Ghatkopar E	Mumbai	400077	Maharashtra

In [31]:

```
order = order.iloc[1:]
#deleting the first row
order.head()
```

Out[31]:

	Name	Address 1	Adress 2	Adress 3	City	Pincode	State
1	SURJEET SINGH KADIAN	695-35 JANTA COLONY	0	0	Rohtak	124001	Haryana
2	LAKSHMI NARAYANAN S	K M B L CEEBROS CENTRE 1ST FLR	39 MONTEITH ROAD EGMORE	0	Chennai	600008	Tamil Nadu
3	TARUN SINHA	C-1052 1ST FLOOR	SUSHANT LOK - I	0	Gurgaon	122001	Haryana
4	AVNISH SRIVASTAVA	Fltno207 208 Dhavalgiri	Neelkanth Vihar Kurla Terminus	Pipeline Rd Ghatkopar E	Mumbai	400077	Maharashtra
5	RISHI KHANDELWAL	OM KOTAK MAHINDRA LIFE INSU CO	7TH FLOOR AMBADEP 14 K G MARG	0	New Delhi	110001	Delhi

In [32]:

```
null_counts = order.isnull().sum()
print(null_counts)
```

```
Name         0
Address 1    1
Adress 2     2
Adress 3     0
City         0
Pincode      2
State        0
dtype: int64
```

In [35]:

```
order['Address 1'] = order['Address 1'].fillna("nill")
order['Adress 2'] = order['Adress 2'].fillna("nill")
order['Pincode'] =order['Pincode'].fillna("nill")
order.isnull().sum()
```

Out[35]:

```
Name         0
Address 1    0
Adress 2     0
Adress 3     0
City         0
Pincode      0
State        0
pincode      0
dtype: int64
```

Task:11 Products Review ATS: Review reply Automation Validation Write a python programm to reply review in product page where customers write feedback on Products . Input value link = Yoshops.com .Output = create excel file with web url, Products name, Products Details, products review, customer name and customr email id columns.

In [36]:

```
import pandas as pd
import requests
from bs4 import BeautifulSoup

def scrape_product_reviews(url):
    # Send a request to the website
    response = requests.get(url)

    # Check if the request was successful
    if response.status_code == 200:
        soup = BeautifulSoup(response.content, 'html.parser')

        # Extract the required information from the website using BeautifulSoup
        # Replace the selectors below with the appropriate ones for the specific website
        product_name = soup.select_one('.product-name').text.strip()
        product_details = soup.select_one('.product-details').text.strip()

        reviews = []
        for review in soup.select('.customer-review'):
            customer_name = review.select_one('.customer-name').text.strip()
            customer_email = review.select_one('.customer-email').text.strip()
            review_text = review.select_one('.review-text').text.strip()
            reviews.append({'Customer Name': customer_name,
                            'Customer Email': customer_email,
                            'Review Text': review_text})

        # Create a DataFrame from the collected data
        df = pd.DataFrame(reviews)
        df['Web URL'] = url
        df['Product Name'] = product_name
        df['Product Details'] = product_details
        # Return the DataFrame
        return df
    else:
        print("Failed to retrieve data from the website.")
        return None

if __name__ == "__main__":
    # Input the URL for product reviews
    input_url = "https://www.yoshops.com/products/12345"

    # Scrape the product reviews
    reviews_df = scrape_product_reviews(input_url)

    if reviews_df is not None:
        # Create an Excel file with the scraped data
        output_file = 'product_reviews.xlsx'
        reviews_df.to_excel(output_file, index=False)
        print(f"Product reviews scraped and saved to '{output_file}'.")
```

```
Failed to retrieve data from the website.
```

In [ ]: