Technical Assignment
for
Software Developer C#

Name: Abhijeet Mahto

Language Used: Python

Drive Link:
https://drive.google.com/drive/folders/1oB6dNdrAu-BAwGUQRek6tvc5b3ueEBko?usp=sharing

Tasks

a) Visualize JSON data via HTML table

Write a program that will create a html page, showing a table of employees that are ordered by the total time worked. The table should show the Name, and the Total time worked. Color the table row if the employee worked less than 100 hours. The data for this task must be retrieved from a GET call on following API endpoint:
https://rc-vault-fap-live-1.azurewebsites.net/api/gettimeentries?code=vO17RnE8vuzXzPJo5eaLLjXjmRW07law99QTD90zat9FfOQJKKUcgQ==

b) Visualize JSON data in a PIE Chart

Write a program that will generate a PNG(image) file using the data from the REST endpoint. Pie-chart will be showing a percentage of the total time worked by an employee.

Codes:-

App.py

```
# Technical Assignment for Software Developer CS
# Name : Abhijeet Mahto

from flask import Flask, render_template
import pandas as pd
import datetime
import matplotlib.pyplot as plt
from io import BytesIO
import base64

app =
Flask(__name__,template_folder=r'C:\Users\ababh\Desktop\template
s')

@app.route('/')
def index():
    # Load data into a Pandas DataFrame
    URL = "https://rc-vault-fap-live-
1.azurewebsites.net/api/gettimeentries?code=vO17RnE8vuzXzPJo5ea
LLjXjmRW07law99QTD90zat9FfOQJKKUcgQ=="
    df = pd.read_json(URL)

        # Question 1 : Visualize JSON data via HTML table
```

```
    df['start_time'] = df['StarTimeUtc'].apply(lambda x:
datetime.datetime.fromisoformat(x))
    df['end_time'] = df['EndTimeUtc'].apply(lambda x:
datetime.datetime.fromisoformat(x))
    df['time_of_work'] = abs(df['end_time'] - df['start_time'])
    df['work_hours'] = (df['time_of_work']) / pd.Timedelta(hours=1)
    df_new=df[['EmployeeName','time_of_work', 'work_hours']]


df_total=df_new.groupby(df["EmployeeName"])["work_hours"].sum(
)
    df_total=df_total.reset_index(name="work_hours")
    #df_total['highlighted']=df_total['work_hours']>100
    def color_row(df):
        if df['work_hours']< 100:
            return ['background-color: yellow']*len(df)
        else:
            return ['']*len(df)
    df_styled=df_total.style.apply(color_row,axis=1)
    # Create a pie chart of the total work hours by employee
    hours_by_employee =
df_total.groupby('EmployeeName')['work_hours'].sum()


    df_table=df_styled.to_html(classes='table',index=False)
```

Question 2 : Visualize JSON data in a PIE Chart

```
    plt.pie(hours_by_employee.values,
labels=hours_by_employee.index,autopct='%1.1f%%')
    plt.title('Total Work Hours by Employee')



    # Save the pie chart as a PNG image
    buffer = BytesIO()
    plt.savefig(buffer, format='png')
    buffer.seek(0)
    chart_data = base64.b64encode(buffer.read()).decode('utf-8')



    # Render the template with the DataFrame and chart attached as
variables
    return render_template('my_template.html',
chart_data=chart_data,table=df_table)



if __name__ == '__main__':
    app.run(debug=True)
```

my_template.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My Template</title>
    <style>
      .highlight {
        background-color: yellow;
      }
    </style>
  </head>
  <body>
    <h1>Employee Work Hours</h1>
    {{ table | safe}}
    <h2>Total Work Hours by Employee</h2>
    <img src="{{ chart_data }}" alt="Pie Chart">
  </body>
</html>
```

Code Explaination:-

```
from flask import Flask, render_template
import pandas as pd
import datetime
import matplotlib.pyplot as plt
from io import BytesIO
import base64
```

- *The above line of codes are the necessary libraries which are imported.*
- *"Flask" if for creating web application and "render_template" is to render a web page with a template.*
- *"pandas" is used for data manipulation which are fetched from the given API.*
- *"datetime" helps to work with date and time values.*
- *"matplotlib" is used to create visualization. As I have created Pie-Chart.*
- *"BytesIO" and "base64" is used to encode the image data.*

```
URL = "https://rc-vault-fap-live-1.azurewebsites.net/api/gettimeentries?code=vO17RnE8vuzXzPJo5eaLLjXjmRW07law99QTD90zat9FfOQJKKUcgQ=="
df = pd.read_json(URL)
```

- *The above line of code is used to load data from a JSON API into a pandas dataframe using read_json(URL) method.*

```
df['start_time'] = df['StarTimeUtc'].apply(lambda x:
datetime.datetime.fromisoformat(x))

df['end_time'] = df['EndTimeUtc'].apply(lambda x:
datetime.datetime.fromisoformat(x))

df['time_of_work'] = abs(df['end_time'] - df['start_time'])

df['work_hours'] = (df['time_of_work']) / pd.Timedelta(hours=1)

df_new=df[['EmployeeName','time_of_work', 'work_hours']]

df_total=df_new.groupby(df["EmployeeName"])["work_hours"].sum(
)

df_total=df_total.reset_index(name="work_hours")
```

- *The above line of codes is used to perform some data manipulation on the DataFrame to extract information of Total Time Work Hours of an individual Employee.*
- *First values are converted into DateTime format and than Total Time Work is calculated.*

```
def color_row(df):
    if df['work_hours']< 100:
        return ['background-color: yellow']*len(df)
    else:
        return ['']*len(df)
df_styled=df_total.style.apply(color_row,axis=1)
```

- *The above line of code is to style the DateFrame table by highlighting the rows where the work hours are less than 100 hrs.*

```
hours_by_employee =
df_total.groupby('EmployeeName')['work_hours'].sum()

plt.pie(hours_by_employee.values,
labels=hours_by_employee.index,autopct='%1.1f%%')

plt.title('Total Work Hours by Employee')
```

- *The above line of codes is creating a pie chart using Matplotlib library to show the Total Work Hours by each Employee.*

```
buffer = BytesIO()

plt.savefig(buffer, format='png')

buffer.seek(0)

chart_data = base64.b64encode(buffer.read()).decode('utf-8')
```

- These line of code is saving the pie chart as a PNG image and encode it with the help of base64.

Snippets:-

Codes:-

App.py

```
# Technical Assignment for Software Developer CS
# Name : Abhijeet Mahto

from flask import Flask, render_template
import pandas as pd
import datetime
import matplotlib.pyplot as plt
from io import BytesIO
import base64

app = Flask(__name__,template_folder=r'C:\Users\ababh\Desktop\templates')

@app.route('/')
def index():
    # Load data into a Pandas DataFrame
    URL = "https://rc-vault-fap-live-1.azurewebsites.net/api/gettimeentries?code=vO17RnE8vuzXzPJo5eaLLjXjmRW07law99QTD90zat9FfOQJKKUcgQ=="
    df = pd.read_json(URL)

    # Question 1 : Visualize JSON data via HTML table

    df['start_time'] = df['StarTimeUtc'].apply(lambda x: datetime.datetime.fromisoformat(x))
    df['end_time'] = df['EndTimeUtc'].apply(lambda x: datetime.datetime.fromisoformat(x))
    df['time_of_work'] = abs(df['end_time'] - df['start_time'])
    df['work_hours'] = (df['time_of_work']) / pd.Timedelta(hours=1)
    df_new=df[['EmployeeName','time_of_work', 'work_hours']]

    df_total=df_new.groupby(df["EmployeeName"])["work_hours"].sum()
    df_total=df_total.reset_index(name="work_hours")
    #df_total['highlighted']=df_total['work_hours']>100



    def color_row(df):
        if df['work_hours']< 100:
            return ['background-color: yellow']*len(df)
        else:
            return ['']*len(df)
    df_styled=df_total.style.apply(color_row,axis=1)
    # Create a pie chart of the total work hours by employee
    hours_by_employee = df_total.groupby('EmployeeName')['work_hours'].sum()

    df_table=df_styled.to_html(classes='table',index=False)
```

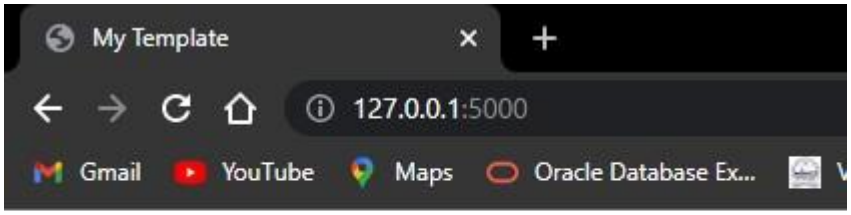
```
    # Question 2 : Visualize JSON data in a PIE Chart

    plt.pie(hours_by_employee.values, labels=hours_by_employee.index,autopct='%1.1f%%')
    plt.title('Total Work Hours by Employee')


    # Save the pie chart as a PNG image
    buffer = BytesIO()
    plt.savefig(buffer, format='png')
    buffer.seek(0)
    chart_data = base64.b64encode(buffer.read()).decode('utf-8')


    # Render the template with the DataFrame and chart attached as variables
    return render_template('my_template.html', chart_data=chart_data,table=df_table)


if __name__ == '__main__':
    app.run(debug=True)
```

My_template.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My Template</title>
    <style>
      .highlight {
        background-color: yellow;
      }
    </style>
  </head>
  <body>
    <h1>Employee Work Hours</h1>
    {{ table | safe}}
    <h2>Total Work Hours by Employee</h2>
    <img src="{{ chart_data }}" alt="Pie Chart">
  </body>
</html>
```

Output:-

Question 1:-

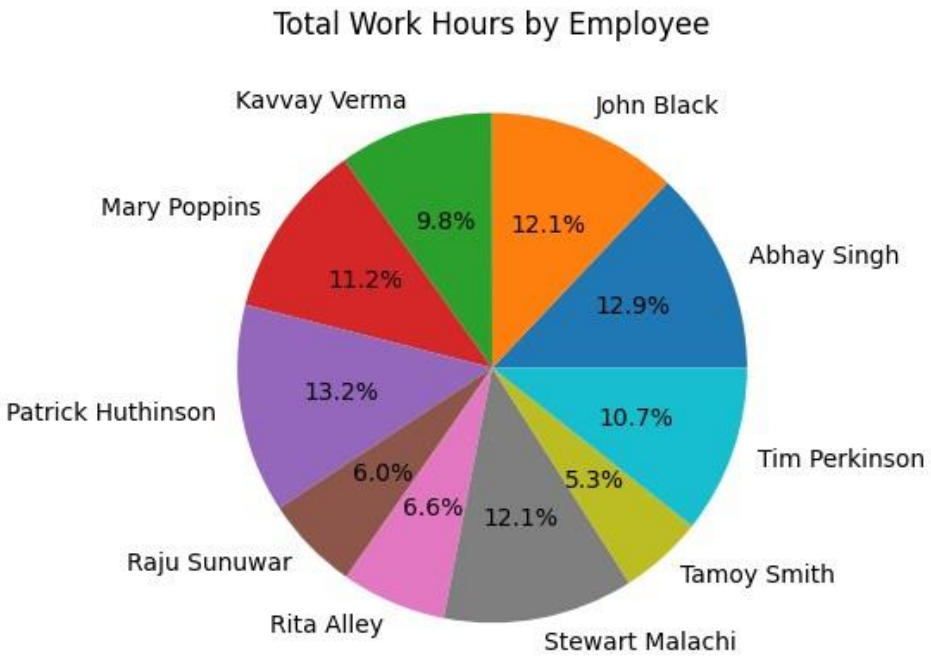


Employee Work Hours

	EmployeeName	work_hours
0	Abhay Singh	221.183333
1	John Black	206.750000
2	Kavvay Verma	167.950000
3	Mary Poppins	190.900000
4	Patrick Huthinson	226.533333
5	Raju Sunuwar	103.150000
6	Rita Alley	113.600000
7	Stewart Malachi	206.633333
8	Tamoy Smith	90.900000
9	Tim Perkinson	182.116667

Question 2:-

Total Work Hours by Employee



Employee Work Hours

	EmployeeName	work_hours
0	Abhay Singh	221.183333
1	John Black	206.750000
2	Kavvay Verma	167.950000
3	Mary Poppins	190.900000
4	Patrick Huthinson	226.533333
5	Raju Sunuwar	103.150000
6	Rita Alley	113.600000
7	Stewart Malachi	206.633333
8	Tamoy Smith	90.900000
9	Tim Perkinson	182.116667

Total Work Hours by Employee