

========================================================================

app.py

========================================================================

import io

import base64

import matplotlib.pyplot as plt

from flask import Flask, render\_template

import pandas as pd

import plotly.express as px

import io

import base64

import matplotlib.pyplot as plt

import plotly.io as pio

import matplotlib

matplotlib.use('Agg')

app = Flask(\_\_name\_\_)

class ImagePlot:

"""

Class for creating and encoding plots using Matplotlib.

This class provides methods to generate plots and convert them

to base64 encoded format for easy storage or transmission.

"""

@staticmethod

def plot\_image(plot\_func):

"""

Generate a plot and return it as a base64 encoded string.

Parameters:

plot\_func (callable): A function that creates the plot.

Returns:

str: A base64 encoded string representing the generated plot.

"""

# Call the function to get the figure

fig = plot\_func()

if isinstance(fig, plt.Figure):

# Handle Matplotlib figures

buf = io.BytesIO()

fig.savefig(buf, format='png')

plt.close(fig)

buf.seek(0)

img\_base64 = base64.b64encode(buf.read()).decode('utf-8')

else:

# Handle Plotly figures

img\_bytes = pio.to\_image(fig, format='png')

img\_base64 = base64.b64encode(img\_bytes).decode('utf-8')

return img\_base64

itemsq1 = [

"Pie Chart for How many students got more than 70 score, between 60 to 70, between 40 to 60 and below 40 in math.",

"Pie Chart for How many students got more than 70 score, between 60 to 70, between 40 to 60 and below 40 in reading.",

"Pie Chart for How many students got more than 70 score, between 60 to 70, between 40 to 60 and below 40 in writing.",

"Bar chart for how many student completed test preparation course and how many student not completed test preparation course.",

"Line chart for math score of 20 to 30 roll nos.",

"Display the count with proper design that how many students parent having bachelor and master degree.",

"In dataset, replace data of lunch for free/reduced to premium and display the count of no of students who choose standard lunch and premium lunch."]

df = pd.read\_csv('./StudentsPerformance.csv')

def plot\_q1():

i = 'math score'

above\_70 = df[df[i] > 70].shape[0]

between\_60\_and\_70 = df[(df[i] > 60) & (df[i] <= 70)].shape[0]

between\_40\_and\_60 = df[(df[i] > 40) & (df[i] <= 60)].shape[0]

below\_40 = df[df[i] <= 40].shape[0]

values = [above\_70,between\_60\_and\_70,between\_40\_and\_60,below\_40]

dist = ["Distinction (Above 70)","First class (Between 70 - 60)","Second class (Between 40 - 60)","Fail (Below 40)"]

# Plotly interactive pie chart

fig = px.pie(

values=values,

names=dist,

title=f'{i} Score Distribution',

hover\_name=dist

)

fig.update\_traces(hovertemplate='%{label}: %{value} students')

return fig

def plot\_q2():

i = 'reading score'

above\_70 = df[df[i] > 70].shape[0]

between\_60\_and\_70 = df[(df[i] > 60) & (df[i] <= 70)].shape[0]

between\_40\_and\_60 = df[(df[i] > 40) & (df[i] <= 60)].shape[0]

below\_40 = df[df[i] <= 40].shape[0]

values = [above\_70,between\_60\_and\_70,between\_40\_and\_60,below\_40]

dist = ["Distinction (Above 70)","First class (Between 70 - 60)","Second class (Between 40 - 60)","Fail (Below 40)"]

# Plotly interactive pie chart

fig = px.pie(

values=values,

names=dist,

title=f'{i} Score Distribution',

hover\_name=dist

)

fig.update\_traces(hovertemplate='%{label}: %{value} students')

return fig

def plot\_q3():

i = 'writing score'

above\_70 = df[df[i] > 70].shape[0]

between\_60\_and\_70 = df[(df[i] > 60) & (df[i] <= 70)].shape[0]

between\_40\_and\_60 = df[(df[i] > 40) & (df[i] <= 60)].shape[0]

below\_40 = df[df[i] <= 40].shape[0]

values = [above\_70,between\_60\_and\_70,between\_40\_and\_60,below\_40]

dist = ["Distinction (Above 70)","First class (Between 70 - 60)","Second class (Between 40 - 60)","Fail (Below 40)"]

# Plotly interactive pie chart

fig = px.pie(

values=values,

names=dist,

title=f'{i} Score Distribution',

hover\_name=dist

)

fig.update\_traces(hovertemplate='%{label}: %{value} students')

return fig

def plot\_q4():

completeTest = df[df['test preparation course'] == "completed"]['Roll No'].count()

pendingTest = df[df['test preparation course'] == "none"]['Roll No'].count()

plt.bar(['Complete Test Preparation','Pending Test Preparation'],[completeTest,pendingTest],color=['purple','indigo'])

plt.title("Parent having complete and pending test preparation")

plt.xlabel("Test")

plt.ylabel("No. Of Test")

return plt.gcf()

def plot\_q5():

x = df[(df['Roll No'] > 20) & (df['Roll No'] < 30)]

plt.plot(x['math score'],x['Roll No'],'o')

plt.plot(x['math score'],x['Roll No'])

plt.title("Line Chart for math score of 20 - 30 roll no")

plt.xlabel("Math Score")

plt.ylabel("Roll No.")

return plt.gcf()

def plot\_q6():

bachelor = df[df['parental level of education'] == "bachelor's degree"]['Roll No'].count()

master = df[df['parental level of education'] == "master's degree"]['Roll No'].count()

plt.bar(['Bachelor\'s degree','Master\'s degree'],[bachelor,master],color=['orange','red'])

plt.title("Parent having bachelor and master")

plt.xlabel("Degree")

plt.ylabel("No. Of Degree")

return plt.gcf()

def plot\_q7():

newDf = df

newDf['lunch'] = newDf['lunch'].str.replace('free/reduced','premium')

standard = newDf[newDf['lunch'] == "standard"]['Roll No'].count()

premium = newDf[newDf['lunch'] == "premium"]['Roll No'].count()

plt.bar(['Premium lunch','Standard lunch'],[premium,standard],color=['teal','green'])

plt.title("Student having premium and standard")

plt.xlabel("Lunch")

plt.ylabel("No. Of student")

return plt.gcf()

@app.route('/')

def index():

return render\_template('welcome.html',itemsq1=itemsq1)

@app.route('/graph/<int:question>')

def graph(question):

if 1 == question:

img = ImagePlot.plot\_image(plot\_q1)

elif 2 == question:

img = ImagePlot.plot\_image(plot\_q2)

elif 3 == question:

img = ImagePlot.plot\_image(plot\_q3)

elif 4 == question:

img = ImagePlot.plot\_image(plot\_q4)

elif 5 == question:

img = ImagePlot.plot\_image(plot\_q5)

elif 6 == question:

img = ImagePlot.plot\_image(plot\_q6)

elif 7 == question:

img = ImagePlot.plot\_image(plot\_q7)

return render\_template('graph.html',title=itemsq1[question-1],src=img)

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

========================================================================

util/footer.html

========================================================================

<footer class="bg-rose-900 text-white mt-10">

<div class="container mx-auto px-4 py-4 text-center">

<h2 class="text-lg font-bold">Assignment-4</h2>

<p class="text-sm">© 2024 Ansh Yadav. All rights reserved.</p>

</div>

</footer>

========================================================================

util/header.html

========================================================================

<header class="bg-rose-700 text-white fixed w-full">

<div class="container mx-auto px-4 py-4 flex justify-between items-center">

<div class="text-lg font-bold">

<a href="{{ url\_for('index') }}" class="hover:text-gray-400">Assignment-4</a>

</div>

<nav>

<ul class="flex space-x-6">

<li>

<a href="{{ url\_for('index') }}" class="hover:text-rose-400 hover:underline hover:underline-offset-4">Home</a>

</li>

{% for i in range(1,8) %}

<li>

<a href="/graph/{{ i }}" class="hover:text-rose-400 hover:underline hover:underline-offset-4">Question {{i}}</a>

</li>

{% endfor %}

</ul>

</nav>

</div>

</header>

========================================================================

util/macros.html

========================================================================

{% macro image(title, src) %}

<div class="mb-10">

<div class="text-2xl font-bold">{{ title }}</div>

<form action="/graph" method="POST">

<input type="hidden" name="title" value="{{ title }}">

<input type="hidden" name="src" value="{{ src }}">

<button type="submit">

<img class="h-auto w-full max-w-4xl" src="data:image/png;base64,{{ src }}" alt="{{ title }}">

</button>

</form>

</div>

{% endmacro %}

========================================================================

graph.html

========================================================================

{% extends "index.html" %}

{% block title %}Graphs {% endblock %}

{% block body %}

<div class=" m-5">

<div class="text-3xl font-bold">{{ title }}</h1>

<img src="data:image/png;base64,{{ src }}">

</div>

</div>

{% endblock %}

========================================================================

index.html

========================================================================

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script src="https://cdn.tailwindcss.com"></script>

<title>{% block title required %}{% endblock %} - Assignment-4</title>

</head>

<body>

{% include 'util/header.html' %}

<div class="px-5 pt-20 min-h-screen">

{% block body required %} {% endblock %}

</div>

{% include 'util/footer.html' %}

</body>

</html>

========================================================================

Welcome.html

========================================================================

{% extends "index.html" %}

{% block title %}Welcome{% endblock %}

{% block body %}

<div>

<div class="text-3xl mt-5 mb-5">List of Charts</div>

<ul class="p-5 bg-rose-100 w-full text-rose-900 rounded-lg font-mono border-2 border-rose-300">

{% for item in itemsq1 %}

<li class="mb-4">{{ loop.index }}) {{ item }}</li>

{% endfor %}

</ul>

</div>

{% endblock %}