# Import required libraries

import pandas as pd

import dash

import dash\_html\_components as html

import dash\_core\_components as dcc

from dash.dependencies import Input, Output

import plotly.express as px

# Read the airline data into pandas dataframe

spacex\_df = pd.read\_csv("spacex\_launch\_dash.csv")

max\_payload = spacex\_df['Payload Mass (kg)'].max()

min\_payload = spacex\_df['Payload Mass (kg)'].min()

launch\_sites=spacex\_df['Launch Site'].unique()#my definition

# Create a dash application

app = dash.Dash(\_\_name\_\_)

# Create an app layout

app.layout = html.Div(children=[html.H1('SpaceX Launch Records Dashboard',

                                        style={'textAlign': 'center', 'color': '#503D36',

                                               'font-size': 40}),

                                # TASK 1: Add a dropdown list to enable Launch Site selection

                                # The default select value is for ALL sites

                                dcc.Dropdown(id='site-dropdown',

                                options=[{'label':x, 'value':x} for x in launch\_sites],

                                value='ALL',

                                placeholder='Select a Launch Site here'),

                                html.Br(),

                                # TASK 2: Add a pie chart to show the total successful launches count for all sites

                                # If a specific launch site was selected, show the Success vs. Failed counts for the site

                                html.Div(dcc.Graph(id='success-pie-chart', figure='fig')),

                                html.Br(),

                                html.P("Payload range (Kg):"),

                                # TASK 3: Add a slider to select payload range

                                #dcc.RangeSlider(id='payload-slider',...)

                                # TASK 4: Add a scatter chart to show the correlation between payload and launch success

                                html.Div(dcc.Graph(id='success-payload-scatter-chart')),

                                ])

# TASK 2:

# Add a callback function for `site-dropdown` as input, `success-pie-chart` as output

@app.callback(

    Output(component\_id='success-pie-chart', component\_property='figure'),

    Input(component\_id='site-dropdown', component\_property='value'))

def pie\_Chart(value):

        if value=='ALL':

            #df\_ALL=spacex\_df.groupby(['Launch Site'])['class'].sum()

            fig=px.pie(spacex\_df, values=spacex\_df.groupby(['Launch Site'])['class'].sum(),

            names=spacex\_df['Launch Site'].unique(), title='Launch Sites vs Success')

        else:

            df\_notALL=spacex\_df[spacex\_df['Launch Site']==value]

            #df\_notALL['class'].value\_counts()

            fig=px.pie(df\_notALL, values=df\_notALL['class'].value\_counts(),

            names=df\_notALL['class'].unique(), title='Launch Site Successes and Failures')

        return fig

# Add a callback function for `site-dropdown` and `payload-slider` as inputs, `success-payload-scatter-chart` as output

# Run the app

if \_\_name\_\_ == '\_\_main\_\_':

    app.run\_server()

@app.callback(

    Output(component\_id='success-payload-scatter-chart', component\_property='figure'),

    Input(component\_id='site-dropdown', component\_property='value'))

    #[Input(component\_id='site-dropdown', component\_property='value'),

    #Input(component\_id='payload-slider', component\_property='value')])

def Scatter\_Chart(value):

        if  value=='ALL':

            fig2=px.scatter(spacex\_df, x='Payload Mass',

            y='class', color='Booster Version Category' ,

            title='Payload Mass (Kg)  vs Success')

        else:

            df\_notALL1=spacex\_df[spacex\_df['Launch Site']==value]

            fig2=px.scatter(x=df\_notALL1['Payload Mass'],

            y=df\_notALL1['class'], color=df\_notALL1['Booster Version Category'],

            title='Payload Mass (Kg)  vs Success')

        return fig2