**# 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()**

**# 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': i, 'value': i} for i in spacex\_df['Launch Site'].unique()**

**] + [{'label': 'All Sites', 'value': 'ALL'}],**

**value='ALL',**

**placeholder="Select a Launch Site",**

**searchable=True),**

**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')),**

**html.Br(),**

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

**# TASK 3: Add a slider to select payload range**

**dcc.RangeSlider(id='payload-slider',**

**min=min\_payload,**

**max=max\_payload,**

**step=1000, # Let's say 1000 kg steps**

**value=[min\_payload, max\_payload],**

**marks={str(payload): str(payload) for payload in range(int(min\_payload), int(max\_payload+1), 5000)}),**

**html.Br(),**

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

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

**])**

**# Callback for TASK 2:**

**@app.callback(**

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

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

**)**

**def update\_graph(selected\_site):**

**if selected\_site == 'ALL':**

**fig = px.pie(spacex\_df, names='Launch Site', values='class')**

**else:**

**filtered\_df = spacex\_df[spacex\_df['Launch Site'] == selected\_site]**

**fig = px.pie(filtered\_df, names='class', title=f'Success launches for site {selected\_site}')**

**return fig**

**# Callback for TASK 4:**

**@app.callback(**

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

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

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

**)**

**def update\_scatter(selected\_site, payload\_range):**

**filtered\_df = spacex\_df[(spacex\_df['Payload Mass (kg)'] >= payload\_range[0]) &**

**(spacex\_df['Payload Mass (kg)'] <= payload\_range[1])]**

**if selected\_site != 'ALL':**

**filtered\_df = filtered\_df[filtered\_df['Launch Site'] == selected\_site]**

**fig = px.scatter(filtered\_df, x='Payload Mass (kg)', y='class',**

**color='Booster Version Category',**

**title=f'Correlation between Payload and Success for {**