|  |
| --- |
| 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': 'All Sites', 'value': 'All Sites'}, |
|  |

|  |
| --- |
| {'label': 'CCAFS LC-40', 'value': 'CCAFS LC-40'}, |
|  |

|  |
| --- |
| {'label': 'VAFB SLC-4E', 'value': 'VAFB SLC-4E'}, |
|  |

|  |
| --- |
| {'label': 'KSC LC-39A', 'value': 'KSC LC-39A'}, |
|  |

|  |
| --- |
| {'label': 'CCAFS SLC-40', 'value': 'CCAFS SLC-40'} |
|  |

|  |
| --- |
| ], |
|  |

|  |
| --- |
| placeholder='Select a Launch Site Here', |
|  |

|  |
| --- |
| value='All Sites', |
|  |

|  |
| --- |
| 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=0, |
|  |

|  |
| --- |
| max=10000, |
|  |

|  |
| --- |
| step=1000, |
|  |

|  |
| --- |
| marks={i: '{}'.format(i) for i in range(0, 10001, 1000)}, |
|  |

|  |
| --- |
| value=[min\_payload, max\_payload]), |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| # 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 get\_pie\_chart(launch\_site): |
|  |

|  |
| --- |
| if launch\_site == 'All Sites': |
|  |

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

|  |
| --- |
| names=spacex\_df.groupby('Launch Site')['Launch Site'].first(), |
|  |

|  |
| --- |
| title='Total Success Launches by Site') |
|  |

|  |
| --- |
| else: |
|  |

|  |
| --- |
| fig = px.pie(values=spacex\_df[spacex\_df['Launch Site']==str(launch\_site)]['class'].value\_counts(normalize=True), |
|  |

|  |
| --- |
| names=spacex\_df['class'].unique(), |
|  |

|  |
| --- |
| title='Total Success Launches for Site {}'.format(launch\_site)) |
|  |

|  |
| --- |
| return(fig) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| # TASK 4: |
|  |

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

|  |
| --- |
| @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 get\_payload\_chart(launch\_site, payload\_mass): |
|  |

|  |
| --- |
| if launch\_site == 'All Sites': |
|  |

|  |
| --- |
| fig = px.scatter(spacex\_df[spacex\_df['Payload Mass (kg)'].between(payload\_mass[0], payload\_mass[1])], |
|  |

|  |
| --- |
| x="Payload Mass (kg)", |
|  |

|  |
| --- |
| y="class", |
|  |

|  |
| --- |
| color="Booster Version Category", |
|  |

|  |
| --- |
| hover\_data=['Launch Site'], |
|  |

|  |
| --- |
| title='Correlation Between Payload and Success for All Sites') |
|  |

|  |
| --- |
| else: |
|  |

|  |
| --- |
| df = spacex\_df[spacex\_df['Launch Site']==str(launch\_site)] |
|  |

|  |
| --- |
| fig = px.scatter(df[df['Payload Mass (kg)'].between(payload\_mass[0], payload\_mass[1])], |
|  |

|  |
| --- |
| x="Payload Mass (kg)", |
|  |

|  |
| --- |
| y="class", |
|  |

|  |
| --- |
| color="Booster Version Category", |
|  |

|  |
| --- |
| hover\_data=['Launch Site'], |
|  |

|  |
| --- |
| title='Correlation Between Payload and Success for Site {}'.format(launch\_site)) |
|  |

|  |
| --- |
| return(fig) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| # Run the app |
|  |

|  |
| --- |
| if \_\_name\_\_ == '\_\_main\_\_': |
|  |

app.run\_server()