

Covid_19\First.py

```
1 import numpy as np
2 import pandas as pd
3 import plotly.graph_objs as go
4 import dash
5 from dash import dcc
6 from dash import html
7 from dash.dependencies import Input, Output
8 import plotly.express as px
9
10 external_stylesheets = [
11     {
12         'href': 'https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css',
13         'rel': 'stylesheet',
14         'integrity': 'sha384 Vko08x4CGs03 + Hhvx8T / Q5PaXtkKtu6ug5TOeNV6-
15         gBiFeWPGFN9MuhOf23Q9Ifjh',
16         'crossorigin': 'anonymous'
17     }
18 ]
19 data = pd.read_csv("state_wise_daily data file IHHPET.csv")
20 Total = data.shape[0]
21 Active = data[data['Status']=='Confirmed'].shape[0]
22 recovered = data[data['Status']=='Recovered'].shape[0]
23 deaths = data[data['Status']=='Deceased'].shape[0]
24
25 options = [
26     {'label': 'All', 'value': 'All'},
27     {'label': 'Hospitalized', 'value': 'Hospitalized'},
28     {'label': 'Recovered', 'value': 'Recovered'},
29     {'label': 'Deceased', 'value': 'Deceased'}
30 ]
31
32 options1 = [
33     {'label': 'All', 'value': 'All'},
34     {'label': 'Mask', 'value': 'Mask'},
35     {'label': 'Sanitizer', 'value': 'Sanitizer'},
36     {'label': 'Oxygen', 'value': 'Oxygen'}
37 ]
38
39 options2 = [
40     {'label': 'All', 'value': 'Status'},
41     {'label': 'Red Zone', 'value': 'Red Zone'},
42     {'label': 'Blue Zone', 'value': 'Blue Zone'},
43     {'label': 'Green Zone', 'value': 'Green Zone'}
44     # {'label': 'Orange Zone', 'value': 'Orange Zone'}
45 ]
46
47 app = dash.Dash(__name__, external_stylesheets=external_stylesheets)
```

```

48
49 app.layout = html.Div([
50
51     html.H1("corona virus pandemic year 2020"),
52     html.Div([
53         html.Div([
54             html.Div([
55                 html.Div([
56                     html.H3("Total Cases", className='text-light'),
57                     html.H4(" = 1791", className="text-light")
58                 ], className='card-body')
59             ], className='card bg-danger')
60         ], className='col-md-3'),
61
62         html.Div([
63             html.Div([
64                 html.Div([
65                     html.H3("Active Cases", className='text-light'),
66                     html.H4(" = 597", className="text-light")
67                 ], className='card-body')
68             ], className='card bg-info')
69         ], className='col-md-3'),
70
71         html.Div([
72             html.Div([
73                 html.Div([
74                     html.H3("Recovered Cases", className='text-light'),
75                     html.H4(" = 597", className="text-light")
76                 ], className='card-body')
77             ], className='card bg-success')
78         ], className='col-md-3'),
79
80         html.Div([
81             html.Div([
82                 html.Div([
83                     html.H3("Total Deaths", className='text-light'),
84                     html.H4(" = 597", className="text-light")
85                 ], className='card-body')
86             ], className='card bg-warning')
87         ], className='col-md-3')
88     ], className='row'),
89
90     html.Div([
91         html.Div([
92             html.Div([
93                 html.Div([
94                     dcc.Dropdown(id='plot-graph', options=options1, value='All'),
95                     dcc.Graph(id='graph')
96                 ], className='card-body')
97             ], className='card bg-info')

```

```

98         ], className='col-md-6'),
99
100     html.Div([
101         html.Div([
102             html.Div([
103                 dcc.Dropdown(id='my_dropdown', options=options2, value='Status', style=
{"width": "100%", "fontSize": "25px"}),
104                 dcc.Graph(id='the_graph')
105             ], className='card-body')
106         ], className='card bg-danger')
107     ], className='col-md-6')
108 ], className='row'),
109
110 html.Div([
111     html.Div([
112         html.Div([
113             html.Div([
114                 dcc.Dropdown(id='picker', options=options, value='All'),
115                 dcc.Graph(id='bar')
116             ], className='card-body')
117         ], className='card bg-warning')
118     ], className='col-md-12')
119 ], className='row')
120 ], className='container')
121
122
123 @app.callback(Output('bar', 'figure'), [Input('picker', 'value')])
124 def update_graph(type):
125
126     if type == 'All':
127         return {'data': [go.Bar(x=data['State'], y=data['Total'])], 'layout':
go.Layout(title="State Total Count : ", plot_bgcolor='orange')}
128
129     if type == 'Hospitalized':
130         return {'data': [go.Bar(x=data['State'], y=data['Total'])], 'layout':
go.Layout(title="Hospitalized", plot_bgcolor='orange')}
131
132     if type == 'Recovered':
133         return {'data': [go.Bar(x=data['State'], y=data['Total'])], 'layout':
go.Layout(title="Recovered", plot_bgcolor='orange')}
134
135     if type == 'Deceased':
136         return {'data': [go.Bar(x=data['State'], y=data['Total'])], 'layout':
go.Layout(title="Deceased", plot_bgcolor='orange')}
137
138
139
140 @app.callback(Output('graph', 'figure'), [Input('plot-graph', 'value')])
141 def generate_graph(type):
142
143     if type == 'All':

```

```
144         return {'data': [go.Line(x=data['Status'], y=data['Total'])], 'layout':
go.Layout(title="Commodities Total Count", plot_bgcolor='pink')}
145
146     if type == 'Mask':
147         return {'data': [go.Line(x=data['Status'], y=data['Total'])], 'layout':
go.Layout(title="Mask", plot_bgcolor='pink')}
148
149     if type == 'Sanitizer':
150         return {'data': [go.Line(x=data['Status'], y=data['Total'])], 'layout':
go.Layout(title="Sanitizer", plot_bgcolor='pink')}
151
152     if type == 'Oxygen':
153         return {'data': [go.Line(x=data['Status'], y=data['Total'])], 'layout':
go.Layout(title="Oxygen", plot_bgcolor='pink')}
154
155
156
157 @app.callback(Output('the_graph', 'figure'), [Input('my_dropdown', 'value')])
158 def generate_graph(my_dropdown):
159
160     chart = px.pie(data_frame=data, names= my_dropdown, hole=0.4)
161     return(chart)
162
163 if __name__ == '__main__':
164     app.run(debug=True)
```