```
import pandas as pd
import json
import glob
import mysql.connector
from sqlalchemy import create_engine
import plotly.express as px
import streamlit as st
import plotly.graph_objects as go
from streamlit_option_menu import option_menu
from plotly.subplots import make_subplots


#Reading csv file using pandas
agg_tran=pd.read_csv("agg_trans.csv")
agg_user=pd.read_csv("agg_users.csv")
map_tran=pd.read_csv("map_trans.csv")
map_user=pd.read_csv("map_users.csv")
top_tran=pd.read_csv("top_trans.csv")
top_user=pd.read_csv("top_users.csv")

#Replacing the state names
agg_tran["state"]=agg_tran["state"].replace({'andaman-&-nicobar-islands': 'Andaman & Nicobar','andhra-pradesh': 'Andhra Pradesh','arunachal-p
'Arunanchal Pradesh','assam': 'Assam','bihar': 'Bihar','chandigarh': 'Chandigarh','chhattisgarh': 'Chhattisgarh','dadra-&-nagar-haveli-&-dama
'Dadara & Nagar Havelli','delhi': 'NCT of Delhi','goa': 'Goa','gujarat': 'Gujarat','haryana': 'Haryana','himachal-pradesh': 'Himachal Pradesh
&-kashmir': 'Jammu & Kashmir','jharkhand': 'Jharkhand','karnataka': 'Karnataka','kerala': 'Kerala','ladakh': 'Ladakh','lakshadweep':
'Lakshadweep','madhya-pradesh': 'Madhya Pradesh','maharashtra': 'Maharashtra','manipur': 'Manipur','meghalaya': 'Meghalaya','mizoram':
'Mizoram','nagaland': 'Nagaland','puducherry': 'Puducherry','punjab': 'Punjab','rajasthan': 'Rajasthan','sikkim': 'Sikkim','tamil-nadu': 'Tam
Nadu','telangana': 'Telangana','tripura': 'Tripura','uttar-pradesh': 'Uttar Pradesh','uttarakhand': 'Uttarakhand','west-bengal': 'West
Bengal','odisha':'Odisha'})
agg_tran["state"]=agg_tran["state"].replace({'andaman-&-nicobar-islands': 'Andaman & Nicobar','andhra-pradesh': 'Andhra Pradesh','arunachal-p
'Arunanchal Pradesh','assam': 'Assam','bihar': 'Bihar','chandigarh': 'Chandigarh','chhattisgarh': 'Chhattisgarh','dadra-&-nagar-haveli-&-dama
'Dadara & Nagar Havelli','delhi': 'NCT of Delhi','goa': 'Goa','gujarat': 'Gujarat','haryana': 'Haryana','himachal-pradesh': 'Himachal Pradesh
&-kashmir': 'Jammu & Kashmir','jharkhand': 'Jharkhand','karnataka': 'Karnataka','kerala': 'Kerala','ladakh': 'Ladakh','lakshadweep':
'Lakshadweep','madhya-pradesh': 'Madhya Pradesh','maharashtra': 'Maharashtra','manipur': 'Manipur','meghalaya': 'Meghalaya','mizoram':
'Mizoram','nagaland': 'Nagaland','puducherry': 'Puducherry','punjab': 'Punjab','rajasthan': 'Rajasthan','sikkim': 'Sikkim','tamil-nadu': 'Tam
Nadu','telangana': 'Telangana','tripura': 'Tripura','uttar-pradesh': 'Uttar Pradesh','uttarakhand': 'Uttarakhand','west-bengal': 'West
Bengal','odisha':'Odisha'})
agg_user["state"]=agg_user["state"].replace({'andaman-&-nicobar-islands': 'Andaman & Nicobar','andhra-pradesh': 'Andhra Pradesh','arunachal-p
'Arunanchal Pradesh','assam': 'Assam','bihar': 'Bihar','chandigarh': 'Chandigarh','chhattisgarh': 'Chhattisgarh','dadra-&-nagar-haveli-&-dama
'Dadara & Nagar Havelli','delhi': 'NCT of Delhi','goa': 'Goa','gujarat': 'Gujarat','haryana': 'Haryana','himachal-pradesh': 'Himachal Pradesh
&-kashmir': 'Jammu & Kashmir','jharkhand': 'Jharkhand','karnataka': 'Karnataka','kerala': 'Kerala','ladakh': 'Ladakh','lakshadweep':
'Lakshadweep','madhya-pradesh': 'Madhya Pradesh','maharashtra': 'Maharashtra','manipur': 'Manipur','meghalaya': 'Meghalaya','mizoram':
'Mizoram','nagaland': 'Nagaland','puducherry': 'Puducherry','punjab': 'Punjab','rajasthan': 'Rajasthan','sikkim': 'Sikkim','tamil-nadu': 'Tam
Nadu','telangana': 'Telangana','tripura': 'Tripura','uttar-pradesh': 'Uttar Pradesh','uttarakhand': 'Uttarakhand','west-bengal': 'West
Bengal','odisha':'Odisha'})
```

```
map_tran["state"]=map_tran["state"].replace({'andaman-&-nicobar-islands': 'Andaman & Nicobar','andhra-pradesh': 'Andhra Pradesh','arunachal-p
'Arunanchal Pradesh','assam': 'Assam','bihar': 'Bihar','chandigarh': 'Chandigarh','chhattisgarh': 'Chhattisgarh','dadra-&-nagar-haveli-&-dama
'Dadara & Nagar Havelli','delhi': 'NCT of Delhi','goa': 'Goa','gujarat': 'Gujarat','haryana': 'Haryana','himachal-pradesh': 'Himachal Pradesh
&-kashmir': 'Jammu & Kashmir','jharkhand': 'Jharkhand','karnataka': 'Karnataka','kerala': 'Kerala','ladakh': 'Ladakh','lakshadweep':
'Lakshadweep','madhya-pradesh': 'Madhya Pradesh','maharashtra': 'Maharashtra','manipur': 'Manipur','meghalaya': 'Meghalaya','mizoram':
'Mizoram','nagaland': 'Nagaland','puducherry': 'Puducherry','punjab': 'Punjab','rajasthan': 'Rajasthan','sikkim': 'Sikkim','tamil-nadu': 'Tam
Nadu','telangana': 'Telangana','tripura': 'Tripura','uttar-pradesh': 'Uttar Pradesh','uttarakhand': 'Uttarakhand','west-bengal': 'West
Bengal','odisha':'Odisha'})
map_user["state"]=map_user["state"].replace({'andaman-&-nicobar-islands': 'Andaman & Nicobar','andhra-pradesh': 'Andhra Pradesh','arunachal-p
'Arunanchal Pradesh','assam': 'Assam','bihar': 'Bihar','chandigarh': 'Chandigarh','chhattisgarh': 'Chhattisgarh','dadra-&-nagar-haveli-&-dama
'Dadara & Nagar Havelli','delhi': 'NCT of Delhi','goa': 'Goa','gujarat': 'Gujarat','haryana': 'Haryana','himachal-pradesh': 'Himachal Pradesh
&-kashmir': 'Jammu & Kashmir','jharkhand': 'Jharkhand','karnataka': 'Karnataka','kerala': 'Kerala','ladakh': 'Ladakh','lakshadweep':
'Lakshadweep','madhya-pradesh': 'Madhya Pradesh','maharashtra': 'Maharashtra','manipur': 'Manipur','meghalaya': 'Meghalaya','mizoram':
'Mizoram','nagaland': 'Nagaland','puducherry': 'Puducherry','punjab': 'Punjab','rajasthan': 'Rajasthan','sikkim': 'Sikkim','tamil-nadu': 'Tam
Nadu','telangana': 'Telangana','tripura': 'Tripura','uttar-pradesh': 'Uttar Pradesh','uttarakhand': 'Uttarakhand','west-bengal': 'West
Bengal','odisha':'Odisha'})
top_tran["state"]=top_tran["state"].replace({'andaman-&-nicobar-islands': 'Andaman & Nicobar','andhra-pradesh': 'Andhra Pradesh','arunachal-p
'Arunanchal Pradesh','assam': 'Assam','bihar': 'Bihar','chandigarh': 'Chandigarh','chhattisgarh': 'Chhattisgarh','dadra-&-nagar-haveli-&-dama
'Dadara & Nagar Havelli','delhi': 'NCT of Delhi','goa': 'Goa','gujarat': 'Gujarat','haryana': 'Haryana','himachal-pradesh': 'Himachal Pradesh
&-kashmir': 'Jammu & Kashmir','jharkhand': 'Jharkhand','karnataka': 'Karnataka','kerala': 'Kerala','ladakh': 'Ladakh','lakshadweep':
'Lakshadweep','madhya-pradesh': 'Madhya Pradesh','maharashtra': 'Maharashtra','manipur': 'Manipur','meghalaya': 'Meghalaya','mizoram':
'Mizoram','nagaland': 'Nagaland','puducherry': 'Puducherry','punjab': 'Punjab','rajasthan': 'Rajasthan','sikkim': 'Sikkim','tamil-nadu': 'Tam
Nadu','telangana': 'Telangana','tripura': 'Tripura','uttar-pradesh': 'Uttar Pradesh','uttarakhand': 'Uttarakhand','west-bengal': 'West
Bengal','odisha':'Odisha'})
top_user["state"]=top_user["state"].replace({'andaman-&-nicobar-islands': 'Andaman & Nicobar','andhra-pradesh': 'Andhra Pradesh','arunachal-p
'Arunanchal Pradesh','assam': 'Assam','bihar': 'Bihar','chandigarh': 'Chandigarh','chhattisgarh': 'Chhattisgarh','dadra-&-nagar-haveli-&-dama
'Dadara & Nagar Havelli','delhi': 'NCT of Delhi','goa': 'Goa','gujarat': 'Gujarat','haryana': 'Haryana','himachal-pradesh': 'Himachal Pradesh
&-kashmir': 'Jammu & Kashmir','jharkhand': 'Jharkhand','karnataka': 'Karnataka','kerala': 'Kerala','ladakh': 'Ladakh','lakshadweep':
'Lakshadweep','madhya-pradesh': 'Madhya Pradesh','maharashtra': 'Maharashtra','manipur': 'Manipur','meghalaya': 'Meghalaya','mizoram':
'Mizoram','nagaland': 'Nagaland','puducherry': 'Puducherry','punjab': 'Punjab','rajasthan': 'Rajasthan','sikkim': 'Sikkim','tamil-nadu': 'Tam
Nadu','telangana': 'Telangana','tripura': 'Tripura','uttar-pradesh': 'Uttar Pradesh','uttarakhand': 'Uttarakhand','west-bengal': 'West
Bengal','odisha':'Odisha'})

#function block for transactions
def tran(menu1,menu2):
       a=agg_tran[(agg_tran.year == menu1) & (agg_tran.quarter == menu2)]
       a=a.groupby(["state","year","quarter"]).sum()
       a.reset_index(inplace=True)
       return a

#function block for users
def user(menu4,menu5,menu6):
       b=agg_user[(agg_user.year == menu4) &(agg_user.quarter == menu5)]
       b.reset_index(inplace=True)
       return b

#function block for state wise transction analysis
```

```
def aggTrans(menu7,menu8,menu9,menu10):
    c= agg_tran[(agg_tran.state ==menu7) & (agg_tran.year == menu8) & (agg_tran.quarter == menu9)]
    c.reset_index(inplace = True)
    return c

#function block for state wise users analysis
def aggUser(menu11,menu12,menu13,menu14):
    d= agg_user[(agg_user.state ==menu11) & (agg_user.year == menu12) & (agg_user.quarter == menu13)]
    d.reset_index(inplace = True)
    return d


#background
st.markdown(
        f"""
        <style>
        .stApp {{
            background-image: url("https://i.pinimg.com/originals/74/8c/28/748c28cd15f309f6ae3895f6828861f9.jpg");
            background-attachment: fixed;
            background-size: cover
        }}
        </style>
        """,
        unsafe_allow_html=True

       )
#display
st.title(":blue[PhonePe Pulse Dashboard]")
with st.sidebar:
         selected=option_menu(
              menu_title="ALL INDIA",
              options=["Transactions","Users","State wise Transaction Analysis","State wise User Analysis","Top10 Transaction state wise","T
Transaction district wise","Top 10 Transcation pincode wise","TOP 10 Users Registered state wise","TOP 10 Users Registered district wise","TO
Registered pincode wise"],
              icons=["cash","emoji-smile","cash-stack","emoji-sunglasses","chevron-bar-up","caret-up","chevron-bar-up","caret-up","chevron-b
up","caret-up"],
              orientation = "vertical",
       )
         #Transaction code
         if selected=="Transactions":
              menu1 =st.selectbox("select a year",(2018,2019,2020,2021,2022))
              menu2=st.selectbox("Select a quarter", ("Q1","Q2","Q3","Q4"))
```

```
                menu3=st.selectbox(
                      'Transaction count or Transaction amount',("transaction_count","total_amount"))

            #Users code
            if selected=="Users":
                menu4 =st.selectbox("select a year",(2018,2019,2020,2021,2022))
                menu5=st.selectbox("Select a quarter", ("Q1","Q2","Q3","Q4"))
                menu6=st.selectbox(
                      'Registerted users'or'apps opened',("registered_users","apps_opened"))

            #State wise Transaction Analysis
            if selected =="State wise Transaction Analysis":
                  menu7 = st.selectbox(
                            'State state for your choice',
                   ('Andaman & Nicobar','Andhra Pradesh','Arunanchal Pradesh','Assam','Bihar','Chandigarh','Chhattisgarh','Dadara & Nagar
Havelli','Jammu & Kashmir','Jharkhand','Karnataka','Kerala','Ladakh','Lakshadweep','Madhya
Pradesh','Maharashtra','Manipur','Meghalaya','Mizoram','Nagaland','odisha','Puducherry','Punjab','Rajasthan','Sikkim','Tamil
Nadu','Telangana','Tripura','Uttar Pradesh','Uttarakhand','West Bengal'))
                  menu8 = st.selectbox(
                          'select a year',(2018, 2019, 2020, 2021, 2022))
                  menu9= st.selectbox(
                           'select a quarter',("Q1", "Q2", "Q3", "Q4"))
                  menu10= st.selectbox(
                            'Transaction count or amount',("transaction_count","total_amount"))

            #State wise User Analysis
            if selected =="State wise User Analysis":
                  menu11 = st.selectbox(
                            'State state for your choice',
                   ('Andaman & Nicobar','Andhra Pradesh','Arunanchal Pradesh','Assam','Bihar','Chandigarh','Chhattisgarh','Dadara & Nagar
Havelli','Jammu & Kashmir','Jharkhand','Karnataka','Kerala','Ladakh','Lakshadweep','Madhya
Pradesh','Maharashtra','Manipur','Meghalaya','Mizoram','Nagaland','odisha','Puducherry','Punjab','Rajasthan','Sikkim','Tamil
Nadu','Telangana','Tripura','Uttar Pradesh','Uttarakhand','West Bengal'))
                  menu12= st.selectbox(
                          'select a year',(2018, 2019, 2020, 2021, 2022))
                  menu13= st.selectbox(
                           'select a quarter',("Q1", "Q2", "Q3", "Q4"))
                  menu14= st.selectbox(
                            'Registerted users'or'apps opened',("registered_users","apps_opened"))

#code for transactions  map
if selected =="Transactions":
```

```
            a=tran(menu1,menu2)
            if st.sidebar.button("show"):
                    with st.spinner():
                      fig = px.choropleth(
                       a,

    geojson="https://gist.githubusercontent.com/jbrobst/56c13bbbf9d97d187fea01ca62ea5112/raw/e388c4cae20aa53cb5090210a42ebb9b765c0a36/india_state
                       featureidkey='properties.ST_NM',
                       locations='state',
                       color=menu3,
                       color_continuous_scale='ylorbr'
                      )
                      fig.update_geos(fitbounds="locations", visible=False)
                      st.write("Transactions")
                      st.write(fig)

    #code for map users
    if selected =="Users":
              b=user(menu4,menu5,menu6)
              if st.sidebar.button("show"):
                      with st.spinner():
                        fig = px.choropleth(
                             b,

    geojson="https://gist.githubusercontent.com/jbrobst/56c13bbbf9d97d187fea01ca62ea5112/raw/e388c4cae20aa53cb5090210a42ebb9b765c0a36/india_state
                       featureidkey='properties.ST_NM',
                       locations='state',
                       color=menu6,
                       color_continuous_scale='ylorbr'
                        )
                      fig.update_geos(fitbounds="locations", visible=False)
                      st.write("Users")
                      st.write(fig)
    if selected=="State wise Transaction Analysis":
              c= aggTrans(menu7,menu8,menu9,menu10)
              if st.sidebar.button("show"):
                   fig = px.choropleth(
                    c,

    geojson="https://gist.githubusercontent.com/jbrobst/56c13bbbf9d97d187fea01ca62ea5112/raw/e388c4cae20aa53cb5090210a42ebb9b765c0a36/india_state
                    featureidkey='properties.ST_NM',
                    locations='state',
```

```
                    color=menu10,
                    color_continuous_scale='viridis'
                   )
                   fig.update_geos(fitbounds="locations", visible=False)

                   st.write("total transaction")
                   st.write(fig)

   if selected=="State wise User Analysis":
             d= aggUser(menu11,menu12,menu13,menu14)
             if st.sidebar.button("show"):
                  fig = px.choropleth(
                   d,

   geojson="https://gist.githubusercontent.com/jbrobst/56c13bbbf9d97d187fea01ca62ea5112/raw/e388c4cae20aa53cb5090210a42ebb9b765c0a36/india_state
                   featureidkey='properties.ST_NM',
                   locations='state',
                   color=menu14,
                   color_continuous_scale='viridis'
                   )
                   fig.update_geos(fitbounds="locations", visible=False)

                   st.write("total transaction")
                   st.write(fig)

   #TOP 10 Transaction statewise
   if selected =="Top10 Transaction state wise":
                 z= agg_tran.groupby(["state"]).sum()
                 z.reset_index(inplace = True)
                 df=z.sort_values(['transaction_count'], ascending=[False]).head(10)
                 df = df.reset_index(drop=True)
                 df.to_csv("tran_top_10",index=True)
                 fig = px.pie(df, values='transaction_count', names='state', title='Top 10 Transactions statewise',
                 color_discrete_sequence=['#00cc00', '#0099ff', '#ffcc00', '#cc33ff', '#ff6666'])
                  #adjust chart margins and padding
                 fig.update_layout(margin=dict(t=0, b=0, l=0, r=0), plot_bgcolor='rgba(0,0,0,0)')
                 st.write("Top10 Transaction state wise")
                 # Display the chart
                 st.plotly_chart(fig, use_container_width=True)

   #TOP 10 Transaction districtwise
```

```
if selected=="Top10 Transaction district wise":
        y= map_tran.groupby(["district_name"]).sum()
        y.reset_index(inplace = True)
        df1=y.sort_values(['transaction_count'], ascending=[False]).head(10)
        df1 = df1.reset_index(drop=True)
        df1.to_csv("tran_map_top_10",index=True)
        fig1 = px.pie(df1, values='transaction_count', names='district_name', title='Top 10 Transactions  districtwise',
              color_discrete_sequence=['#00cc00', '#0099ff', '#ffcc00', '#cc33ff', '#ff6666'])
         #adjust chart margins and padding
        fig1.update_layout(margin=dict(t=0, b=0, l=0, r=0), plot_bgcolor='rgba(0,0,0,0)')
        st.write("Top10 Transaction district wise")
           # Display the chart
        st.plotly_chart(fig1, use_container_width=True)

#TOP 10 Transaction pincodewise
if selected=="Top 10 Transcation pincode wise":
        x= top_tran.groupby(["district&pincode"]).sum()
        x.reset_index(inplace = True)
        df2=x.sort_values(['transaction_count'], ascending=[False]).head(10)
        df2 = df2.reset_index(drop=True)
        df2.to_csv("tran_pincode_top_10",index=True)
        fig2 = px.pie(df2, values='transaction_count', names='district&pincode', title='Top 10 Transactions  districtwise',
               color_discrete_sequence=['#00cc00', '#0099ff', '#ffcc00', '#cc33ff', '#ff6666'])
         #adjust chart margins and padding
        fig2.update_layout(margin=dict(t=0, b=0, l=0, r=0), plot_bgcolor='rgba(0,0,0,0)')
        st.write("Top10 Transaction pincode wise")
           # Display the chart
        st.plotly_chart(fig2, use_container_width=True)

#TOP 10 Users Registered state wise
if selected=="TOP 10 Users Registered state wise":
        w= agg_user.groupby(["state"]).sum()
        w.reset_index(inplace = True)
        df3=w.sort_values(['registered_users'], ascending=[False]).head(10)
        df3 = df3.reset_index(drop=True)
        df3.to_csv("user_state_top_10",index=True)
        fig3 = px.pie(df3, values='registered_users', names='state', title='TOP 10 Users Registered statewise',
               color_discrete_sequence=['#00cc00', '#0099ff', '#ffcc00', '#cc33ff', '#ff6666'])
         #adjust chart margins and padding
        fig3.update_layout(margin=dict(t=0, b=0, l=0, r=0), plot_bgcolor='rgba(0,0,0,0)')
        st.write("Top10 Users Registerted state wise")
```

```
            # Display the chart
        st.plotly_chart(fig3, use_container_width=True)

    #TOP 10 Users Registered district wise
    if selected=="TOP 10 Users Registered district wise":
            v= map_user.groupby(["states"]).sum()
            v.reset_index(inplace = True)
            df4=v.sort_values(['registered_users'], ascending=[False]).head(10)
            df4 = df4.reset_index(drop=True)
            df4.to_csv("user_map_top_10",index=True)
            fig4 = px.pie(df4, values='registered_users', names='states', title='TOP 10 Users Registered district wise',
                  color_discrete_sequence=['#00cc00', '#0099ff', '#ffcc00', '#cc33ff', '#ff6666'])
             #adjust chart margins and padding
            fig4.update_layout(margin=dict(t=0, b=0, l=0, r=0), plot_bgcolor='rgba(0,0,0,0)')
            st.write("Top10 Users Registerted district wise")
               # Display the chart
            st.plotly_chart(fig4, use_container_width=True)

    #TOP 10 Users Registered pincode wise
    if selected=="TOP 10 Users Registered pincode wise":
            u= top_user.groupby(["district&pin"]).sum()
            u.reset_index(inplace = True)
            df5=u.sort_values(['registered_users'], ascending=[False]).head(10)
            df5 = df5.reset_index(drop=True)
            df5.to_csv("tran_pincode_top_10",index=True)
            fig5 = px.pie(df5, values='registered_users', names='district&pin', title='TOP 10 Users Registered pincode wise',
                   color_discrete_sequence=['#00cc00', '#0099ff', '#ffcc00', '#cc33ff', '#ff6666'])
             #adjust chart margins and padding
            fig5.update_layout(margin=dict(t=0, b=0, l=0, r=0), plot_bgcolor='rgba(0,0,0,0)')
            st.write("Top10 Users Registerted pincode wise")
    # embed chart in Streamlit app
            st.plotly_chart(fig5, use_container_width=True)
```