```
#pip install streamlit plotly mysql-connector-python
#pip install streamlit
#
import mysql.connector
import pandas as pd
#import psycopg2
import streamlit as st
import PIL
from PIL import Image
from streamlit_option_menu import option_menu
import plotly.express as px
import pandas as pd

import requests

# connect to the database
import mysql.connector

#establishing the connection
conn = mysql.connector.connect(user='root', password='1234',
host='127.0.0.1', database="phonepe_pulse")

# create a cursor object
cursor = conn.cursor()



#with st.sidebar:
SELECT = option_menu(
menu_title = None,
options = ["About","Home","Basic insights","Contact"],
```

```
icons =["bar-chart","house","toggles","at"],
default_index=2,
orientation="horizontal",
styles={"container": {"padding": "0!important", "background-color":
"white","size":"cover", "width": "100%"},
"icon": {"color": "black", "font-size": "20px"},
"nav-link": {"font-size": "20px", "text-align": "center", "margin": "-2px",
"--hover-color": "#6F36AD"},
"nav-link-selected": {"background-color": "#6F36AD"}})



#----------------------Basic Insights -----------------#



if SELECT == "Basic insights":
st.title("BASIC INSIGHTS")
st.write("----")
st.subheader("Let's know some basic insights about the data")
options = ["--select--",
"Top 10 states based on year and amount of transaction",
"List 10 states based on type and amount of transaction",
"Top 5 Transaction_Type based on Transaction_Amount",
"Top 10 Registered-users based on States and District",
"Top 10 Districts based on states and Count of transaction",
"List 10 Districts based on states and amount of transaction",
"List 10 Transaction_Count based on Districts and states",
"Top 10 RegisteredUsers based on states and District"]

#1

select = st.selectbox("Select the option",options)
if select=="Top 10 states based on year and amount of transaction":
cursor.execute("SELECT DISTINCT States, Transaction_Year,
SUM(Transaction_Amount) AS Total_Transaction_Amount FROM top_tran GROUP BY
States, Transaction_Year ORDER BY Total_Transaction_Amount DESC LIMIT 10");
```

```
df = pd.DataFrame(cursor.fetchall(), columns=['States','Transaction_Year',
'Transaction_Amount'])
col1,col2 = st.columns(2)
with col1:
st.write(df)
with col2:
st.title("Top 10 states and amount of transaction")
st.bar_chart(data=df,x="Transaction_Amount",y="States")

#2

elif select=="List 10 states based on type and amount of transaction":
cursor.execute("SELECT DISTINCT States, SUM(Transaction_Count) as Total
FROM top_tran GROUP BY States ORDER BY Total ASC LIMIT 10");
df = pd.DataFrame(cursor.fetchall(),columns=['States','Total_Transaction'])
col1,col2 = st.columns(2)
with col1:
st.write(df)
with col2:
st.title("List 10 states based on type and amount of transaction")
st.bar_chart(data=df,x="Total_Transaction",y="States")

#3

elif select=="Top 5 Transaction_Type based on Transaction_Amount":
cursor.execute("SELECT DISTINCT Transaction_Type, SUM(Transaction_Amount)
AS Amount FROM agg_user GROUP BY Transaction_Type ORDER BY Amount DESC
LIMIT 5");
df =
pd.DataFrame(cursor.fetchall(),columns=['Transaction_Type','Transaction_Amo
unt '])
col1,col2 = st.columns(2)
with col1:
st.write(df)
with col2:
st.title("Top 5 Transaction_Type based on Transaction_Amount")
st.bar_chart(data=df,x="Transaction_Type",y="Amount")
```

#4

```
elif select=="Top 10 Registered-users based on States and District":
cursor.execute("SELECT DISTINCT State, District, SUM(RegisteredUsers) AS
Users FROM top_user GROUP BY State, District ORDER BY Users DESC LIMIT
10");
df =
pd.DataFrame(cursor.fetchall(),columns=['State','District','RegisteredUsers
'])
col1,col2 = st.columns(2)
with col1:
st.write(df)
with col2:
st.title("Top 10 Registered-users based on States and District")
st.bar_chart(data=df,x="State",y="RegisteredUsers")
```

#5

```
elif select=="Top 10 Districts based on states and Count of transaction":
cursor.execute("SELECT DISTINCT States,District,SUM(Transaction_Count) AS
Counts FROM map_tran GROUP BY States,District ORDER BY Counts DESC LIMIT
10");
df =
pd.DataFrame(cursor.fetchall(),columns=['States','District','Transaction_Co
unt'])
col1,col2 = st.columns(2)
with col1:
st.write(df)
with col2:
st.title("Top 10 Districts based on states and Count of transaction")
st.bar_chart(data=df,x="States",y="Transaction_Count")
```

#6

```
elif select=="List 10 Districts based on states and amount of transaction":
```

```
cursor.execute("SELECT DISTINCT
States,Transaction_year,SUM(Transaction_Amount) AS Amount FROM agg_trans
GROUP BY States, Transaction_year ORDER BY Amount ASC LIMIT 10");
df =
pd.DataFrame(cursor.fetchall(),columns=['States','Transaction_year','Transa
ction_Amount'])
col1,col2 = st.columns(2)

with col1:

st.write(df)

with col2:

st.title("Least 10 Districts based on states and amount of transaction")

st.bar_chart(data=df,x="States",y="Transaction_Amount")


#7


elif select=="List 10 Transaction_Count based on Districts and states":

cursor.execute("SELECT DISTINCT States, District, SUM(Transaction_Count) AS
Counts FROM map_tran GROUP BY States,District ORDER BY Counts ASC LIMIT
10");
df =
pd.DataFrame(cursor.fetchall(),columns=['States','District','Transaction_Co
unt'])
col1,col2 = st.columns(2)

with col1:

st.write(df)

with col2:

st.title("List 10 Transaction_Count based on Districts and states")

st.bar_chart(data=df,x="States",y="Transaction_Count")


#8


elif select=="Top 10 RegisteredUsers based on states and District":

cursor.execute("SELECT DISTINCT States,District, SUM(RegisteredUsers) AS
Users FROM map_user GROUP BY States,District ORDER BY Users DESC LIMIT
10");
df = pd.DataFrame(cursor.fetchall(),columns =
['States','District','RegisteredUsers'])
col1,col2 = st.columns(2)

with col1:
```

```
st.write(df)
with col2:
st.title("Top 10 RegisteredUsers based on states and District")
st.bar_chart(data=df,x="States",y="RegisteredUsers")



#----------------Home----------------------#
cursor = conn.cursor()

# execute a SELECT statement
cursor.execute("SELECT * FROM agg_trans")

# fetch all rows
rows = cursor.fetchall()

if SELECT == "Home":
col1,col2, = st.columns(2)
col1.image(Image.open("C:/Users/omkar/Downloads/phonepe
photo/phonepe.png"),width = 500)
with col1:
st.subheader("PhonePe is an Indian digital payments and financial
technology company headquartered in Bengaluru, Karnataka, India. PhonePe
was founded in December 2015, by Sameer Nigam, Rahul Chari and Burzin
Engineer. The PhonePe app, based on the Unified Payments Interface (UPI),
went live in August 2016. It is owned by Flipkart, a subsidiary of
Walmart.")
st.download_button("DOWNLOAD THE APP NOW",
"https://www.phonepe.com/app-download/")
with col2:
st.video("C:/Users/Bala/Downloads/phonepe photo/upi.mp4")



df = pd.DataFrame(rows, columns=['States', 'Transaction_Year', 'Quarters',
'Transaction_Type', 'Transaction_Count','Transaction_Amount'])
fig = px.choropleth(df, locations="States", scope="asia", color="States",
hover_name="States",
title="Live Geo Visualization of India")
st.plotly_chart(fig)
```