In [91]:
```
!git clone https://github.com/PhonePe/pulse.git
```

```
fatal: destination path 'pulse' already exists and is not an empty directory.
```

In [7]:
```
import os
import json
import pandas as pd
```

In [8]:
```
import streamlit as st
import pandas as pd
import plotly.express as px
import sqlalchemy
```

In [9]:
```
import os
```

In [95]:
```
path = "C:/Users/VIJAY/Desktop/pulse/data/aggregated/transaction/country/india/state/"
user_state_list = os.listdir(path)
#To get the list of states in India

# This is to create a dataframe

clm = {'State': [], 'Year': [], 'Quarter': [], 'Transaction_type': [],
       'Transaction_count': [], 'Transaction_amount': []}
for i in user_state_list:
    p_i = path+i+"/"
    Agg_yr = os.listdir(p_i)
    for j in Agg_yr:
        p_j = p_i+j+"/"
        Agg_yr_list = os.listdir(p_j)
        for k in Agg_yr_list:
            p_k = p_j+k
            Data = open(p_k, 'r')
            D = json.load(Data)
            try:
                for z in D['data']['transactionData']:
                    Name = z['name']
                    count = z['paymentInstruments'][0]['count']
                    amount = z['paymentInstruments'][0]['amount']
                    clm['Transaction_type'].append(Name)
                    clm['Transaction_count'].append(count)
                    clm['Transaction_amount'].append(amount)
                    clm['State'].append(i)
                    clm['Year'].append(j)
                    clm['Quarter'].append(int(k.strip('.json')))
            except:
                pass
Agg_Trans = pd.DataFrame(clm)
Agg_Trans.to_csv('Agg_trans.csv')

#<---------------------------------------------------------------------------------

path = "C:/Users/VIJAY/Desktop/pulse/data/aggregated/user/country/india/state/"
user_state_list = os.listdir(path)
#To get the list of states in India

# This is to create a dataframe
```

```
clm = {'State': [], 'Year': [], 'Quarter': [], 'Brand': [],
    'Brand_count': [], 'Brand_percentage': []}
for i in user_state_list:
    p_i = path+i+"/"
    year = os.listdir(p_i)
    for j in year:
        p_j = p_i+j+"/"
        file = os.listdir(p_j)
        for k in file:
            p_k = p_j+k
            Data = open(p_k, 'r')
            D = json.load(Data)
            try:
                for z in D['data']["usersByDevice"]:

                    brand = z['brand']

                    brand_count = z['count']
                    brand_percentage = z["percentage"]
                    clm['Brand'].append(brand)
                    clm['Brand_count'].append(brand_count)
                    clm['Brand_percentage'].append(brand_percentage)
                    clm['State'].append(i)
                    clm['Year'].append(j)
                    clm['Quarter'].append(int(k.strip('.json')))
            except:
                pass

user_by_device = pd.DataFrame(clm)
user_by_device.to_csv('user_by_device.csv')

#<----------------------------------------------------------------------------------

path = "C:/Users/VIJAY/Desktop/pulse/data/map/transaction/hover/country/india/state/"
state_list = os.listdir(path)
# To get the list of states in India

# This is to create a dataframe

clm = {'State': [], 'Year': [], 'Quarter': [], 'District': [],
    'Transaction_count': [], 'Transaction_amount': []}
for i in state_list:
    p_i = path+i+"/"
    year = os.listdir(p_i)
    for j in year:
        p_j = p_i+j+"/"
        file = os.listdir(p_j)
        for k in file:
            p_k = p_j+k
            Data = open(p_k, 'r')
            D = json.load(Data)
            try:
                for z in D['data']["hoverDataList"]:
                    district = z['name']
                    transaction_count = z['metric'][0]['count']
                    transaction_amount = z['metric'][0]['amount']
                    clm['District'].append(district)
                    clm['Transaction_count'].append(transaction_count)
                    clm['Transaction_amount'].append(transaction_amount)
                    clm['State'].append(i)
                    clm['Year'].append(j)
                    clm['Quarter'].append(int(k.strip('.json')))

            except:
                pass


map_transaction = pd.DataFrame(clm)
map_transaction.to_csv('district_map_transaction.csv')

#<-----------------------------------------------------------------------------------

path = "C:/Users/VIJAY/Desktop/pulse/data/map/user/hover/country/india/state/"
state_list = os.listdir(path)

# This is to create a dataframe

clm = {'State': [], 'Year': [], 'Quarter': [], 'District': [],
    'Registered_user': [], 'App_opening': []}
for i in state_list:
    p_i = path+i+"/"
    year = os.listdir(p_i)
    for j in year:
        p_j = p_i+j+"/"
        file = os.listdir(p_j)
        for k in file:
            p_k = p_j+k
            Data = open(p_k, 'r')
            D = json.load(Data)
            try:
                for z in D['data']["hoverData"]:
                    district = z
                    registered_user =  D['data']["hoverData"][z]["registeredUsers"]
                    app_opening = D['data']["hoverData"][z]["appOpens"]
                    clm['District'].append(district)
                    clm['Registered_user'].append(registered_user)
                    clm['App_opening'].append(app_opening)
                    clm['State'].append(i)
                    clm['Year'].append(j)
                    clm['Quarter'].append(int(k.strip('.json')))
            except:
                pass
district_registering = pd.DataFrame(clm)
district_registering.to_csv('district_registering_map.csv')

#<-----------------------------------------------------------------------------------
#latitude and longitude states table
Longitude_Latitude_State_Table = pd.read_csv(r"C:\Users\VIJAY\Desktop\phonepe data vis
Longitude_Latitude_State_Table.to_csv('Longitude_Latitude_State_Table.csv')

#<-----------------------------------------------------------------------------------

#Data mof Districts longitude Lattitude
Districts_Longitude_Latitude = pd.read_csv(r"C:\Users\VIJAY\Desktop\phonepe data visua
Districts_Longitude_Latitude.to_csv('Districts_Longitude_Latitude.csv')
```

In [29]:
```
Agg_Trans= pd.read_csv('Agg_trans.csv')
user_by_device= pd.read_csv('user_by_device.csv')
map_transaction= pd.read_csv('district_map_transaction.csv')
```

```
district_registering= pd.read_csv('district_registering_map.csv')
Longitude_Latitude_State_Table= pd.read_csv('Longitude_Latitude_State_Table.csv')
Districts_Longitude_Latitude= pd.read_csv('Districts_Longitude_Latitude.csv')
```

In [27]:
```
import mysql.connector as sql
mydb = sql.connect(
  host="localhost",
  user="root",
  password="bawadharani",
  autocommit=True
  #database='e55'
)
print(mydb)
mycursor = mydb.cursor(buffered=True)
```

```
<mysql.connector.connection.MySQLConnection object at 0x000001C487F513A0>
```

In [28]:
```
#mycursor.execute("CREATE DATABASE phone_pe")
mycursor.execute("USE phone_pe")
```

In [196…
```
table_zero = '''CREATE TABLE Agg_Transaction_Table (
                    id INT AUTO_INCREMENT PRIMARY KEY,
                    State VARCHAR(255) NOT NULL,
                    Year VARCHAR(255) NOT NULL,
                    Quarter INT NOT NULL,
                    Transaction_type VARCHAR(255) NOT NULL,
                    Transaction_count INT NOT NULL,
                    Transaction_amount FLOAT NOT NULL
                  )'''

# Execute the table schema
mycursor.execute(table_zero)

# Commit the changes and close the connection
mydb.commit()
#mycursor.close()
#mydb.close()

# Load CSV file into a pandas DataFrame
df = pd.read_csv('Agg_trans.csv')

# Insert data into MySQL table
for row in df.itertuples():
    query = "INSERT INTO agg_transaction_table (State, Year, Quarter, Transaction_type
    values = (row.State, row.Year, row.Quarter, row.Transaction_type, row.Transaction_
    mycursor.execute(query, values)

# Commit changes and close connection
mydb.commit()
mycursor.close()
mydb.close()
print("Data loaded successfully")
```

```
Data loaded successfully
```

In [171…
```
# Define the table schema
table_one = '''CREATE TABLE user_by_device (
                    id INT AUTO_INCREMENT PRIMARY KEY,
                    State VARCHAR(255) NOT NULL,
                    Year VARCHAR(255) NOT NULL,
                    Quarter INT NOT NULL,
                    Brand VARCHAR(255) NOT NULL,
                    Brand_count INT NOT NULL,
                    Brand_percentage FLOAT NOT NULL
                  )'''

# Execute the table schema
mycursor.execute(table_one)

# Commit the changes and close the connection
mydb.commit()
#mycursor.close()
#mydb.close()

# Load CSV file into a pandas DataFrame
df = pd.read_csv('user_by_device.csv')

# Insert data into MySQL table
for row in df.itertuples():
    query = "INSERT INTO user_by_device (State, Year, Quarter, Brand, Brand_count, Bra
    values = (row.State, row.Year, row.Quarter, row.Brand, row.Brand_count, row.Brand_
    mycursor.execute(query, values)

# Commit changes and close connection
mydb.commit()
mycursor.close()
mydb.close()
print("Data loaded successfully")
```

```
Data loaded successfully
```

In [177…

```
# Define the table schema
table_two = '''CREATE TABLE district_map_transaction_table (
                    id INT AUTO_INCREMENT PRIMARY KEY,
                    State VARCHAR(255) NOT NULL,
                    Year VARCHAR(255) NOT NULL,
                    Quarter INT NOT NULL,
                    District VARCHAR(255) NOT NULL,
                    Transaction_count INT NOT NULL,
                    Transaction_amount FLOAT NOT NULL
                  )'''

# Execute the table schema
mycursor.execute(table_two)

# Commit the changes and close the connection
mydb.commit()
#mycursor.close()
#mydb.close()

# Load CSV file into a pandas DataFrame
df = pd.read_csv('district_map_transaction.csv')

# Insert data into MySQL table
for row in df.itertuples():
    query = "INSERT INTO district_map_transaction_table (State, Year, Quarter, Distric
    values = (row.State, row.Year, row.Quarter, row.District, row.Transaction_count, r
    mycursor.execute(query, values)

# Commit changes and close connection
```

```
mydb.commit()
mycursor.close()
mydb.close()
print("Data loaded successfully")
```

```
Data loaded successfully
```

In [13]:
```
# Define the table schema
table_two = '''CREATE TABLE district_map_registering_table (
                    id INT AUTO_INCREMENT PRIMARY KEY,
                    State VARCHAR(255) NOT NULL,
                    Year INT NOT NULL,
                    Quarter INT NOT NULL,
                    District VARCHAR(255) NOT NULL,
                    Registered_user INT NOT NULL,
                    App_opening INT NOT NULL
                  )'''

# Execute the table schema
mycursor.execute(table_two)

# Commit the changes and close the connection
mydb.commit()
#mycursor.close()
#mydb.close()

# Load CSV file into a pandas DataFrame
df = pd.read_csv('district_registering_map.csv')

# Insert data into MySQL table
for row in df.itertuples():
    query = "INSERT INTO district_map_registering_table (State, Year, Quarter, Distric
    values = (row.State, row.Year, row.Quarter, row.District, row.Registered_user, row
    mycursor.execute(query, values)

# Commit changes and close connection
mydb.commit()
mycursor.close()
mydb.close()
print("Data loaded successfully")
```

```
Data loaded successfully
```

In [21]:
```
# Define the table schema
table_three = '''CREATE TABLE Longitude_Latitude_State_Table (
                      id INT AUTO_INCREMENT PRIMARY KEY,
                      code VARCHAR(10) NOT NULL,
                      Latitude FLOAT NOT NULL,
                      Longitude FLOAT NOT NULL,
                      state VARCHAR(255) NOT NULL
                    )'''

# Execute the table schema
mycursor.execute(table_three)

# Commit the changes and close the connection
mydb.commit()

# Load CSV file into a pandas DataFrame
df = pd.read_csv('Longitude_Latitude_State_Table.csv')
```

```
# Insert data into MySQL table
for row in df.itertuples():
    query = "INSERT INTO Longitude_Latitude_State_Table (code, Latitude, Longitude, st
    values = (row.code, row.Latitude, row.Longitude, row.state)
    mycursor.execute(query, values)

# Commit changes and close connection
mydb.commit()
mycursor.close()
mydb.close()
print("Data loaded successfully")
```

```
Data loaded successfully
```

In [35]:

```
# Define the table schema
table_three = '''CREATE TABLE Districts_Longitude_Latitude_table (
                      id INT AUTO_INCREMENT PRIMARY KEY,
                      State VARCHAR(100) NOT NULL,
                      Latitude FLOAT NOT NULL,
                      Longitude FLOAT NOT NULL,
                      District VARCHAR(255) NOT NULL
                    )'''

# Execute the table schema
mycursor.execute(table_three)

# Commit the changes and close the connection
mydb.commit()

# Load CSV file into a pandas DataFrame
df = pd.read_csv('Districts_Longitude_Latitude.csv')

# Insert data into MySQL table
for row in df.itertuples():
    query = "INSERT INTO Districts_Longitude_Latitude_table (District, Latitude, Longi
    values = (row.District, row.Latitude, row.Longitude, row.State)
    mycursor.execute(query, values)

# Commit changes and close connection
mydb.commit()
mycursor.close()
mydb.close()
print("Data loaded successfully")
```

```
Data loaded successfully
```

In [ ]:

In [ ]: