**Input output dataset/screenshots**

**Home.py**

import streamlit as st

import pandas as pd

import plotly.express as px

from streamlit\_option\_menu import option\_menu

from numerize.numerize import numerize

from query import \*

import time

st.set\_page\_config(page\_title="Dashboard", page\_icon="🌍",layout="wide")

st.subheader("🚮Insurance Details Analysis")

st.markdown("##")

result=view\_all\_data()

df=pd.DataFrame(result,columns=["Policy","Expiry","Location","State","Region","Investment","Construction","BusinessType","Earthquake","Flood","Rating","id"])

#side bar images

st.sidebar.image("logo1.png",caption="Online Analysis")

#switcher

st.sidebar.header("Please filter")

region=st.sidebar.multiselect(

    "Select Region",

    options=df["Region"].unique(),

    default=df["Region"].unique(),

)

location=st.sidebar.multiselect(

    "Select Location",

    options=df["Location"].unique(),

    default=df["Location"].unique(),

)

construction=st.sidebar.multiselect(

    "Select Construction",

    options=df["Construction"].unique(),

    default=df["Construction"].unique(),

)

df\_selection=df.query(

    "Region==@region & Location==@location & Construction==@construction"

)

def Home():

    with st.expander("Tabular"):

        showData=st.multiselect('Filter: ',df\_selection.columns,default=[])

        st.write(df\_selection[showData])

    #compute the top analytical

    total\_investment = float(df\_selection['Investment'].sum())

    investment\_mode = float(df\_selection['Investment'].mode())

    investment\_mean = float(df\_selection['Investment'].mean())

    investment\_median = float(df\_selection['Investment'].median())

    rating = float(df\_selection['Rating'].sum())

    total1,total2,total3,total4,total5=st.columns(5,gap='large')

    with total1:

        st.info('Total Investment',icon="📌")

        st.metric(label="sum TZS",value=f"{total\_investment: ,.0f}")

    with total2:

        st.info('Most Frequent',icon="📌")

        st.metric(label="mode TZS",value=f"{investment\_mode:,.0f}")

    with total3:

        st.info('Average',icon="📌")

        st.metric(label="average TZS",value=f"{investment\_mean:,.0f}")

    with total4:

        st.info('Central Earning',icon="📌")

        st.metric(label="median TZS",value=f"{investment\_median:,.0f}")

    with total5:

        st.info('Rating',icon="📌")

        st.metric(label="Rating",value=numerize(rating),help=f""" Total Rating: {rating} """)

    st.markdown("""\_ \_ \_""")

#graphs

def graphs():

    #total\_investment=int(df\_selection["Investment"]).sum()

    #averageRating=int(round(df\_selection["Rating"]).mean(),2)

    #simple bar graph

    investment\_by\_business\_type=(

        df\_selection.groupby(by=["BusinessType"]).count()[["Investment"]].sort\_values(by="Investment")

    )

    fig\_investment=px.bar(

        investment\_by\_business\_type,

        x="Investment",

        y=investment\_by\_business\_type.index,

        orientation="h",

        title="<b> Investment by  Business type </b>",

        color\_discrete\_sequence=["#0083B8"]\*len(investment\_by\_business\_type),

        template="plotly\_white",

    )

    fig\_investment.update\_layout(

    plot\_bgcolor="rgba(0,0,0,0)",

    xaxis=(dict(showgrid=False))

      )

    #simple line graph

    investment\_state=df\_selection.groupby(by=["State"]).count()[["Investment"]]

    fig\_state=px.line(

        investment\_state,

        x=investment\_state.index,

        y="Investment",

        orientation="v",

        title="<b> Investment by  state </b>",

        color\_discrete\_sequence=["#0083b8"]\*len(investment\_state),

        template="plotly\_white",

    )

    fig\_state.update\_layout(

    xaxis=dict(tickmode="linear"),

    plot\_bgcolor="rgba(0,0,0,0)",

    yaxis=(dict(showgrid=False))

      )

    left,right=st.columns(2)

    left.plotly\_chart(fig\_state,use\_container\_width=True)

    right.plotly\_chart(fig\_investment,use\_container\_width=True)

def Progressbar():

    st.markdown("""<style>.stProgress > div > div > div > div { background(to right, #99ff99, #FFFF00)}</style""",unsafe\_allow\_html=True,)

    target=3000000000

    current=df\_selection["Investment"].sum()

    percent=round((current/target\*100))

    mybar=st.progress(0)

    if percent>100:

        st.subheader("Target done !")

    else:

        st.write("You have (",percent, "% ", "of ", (format(target, 'd')), "TZS")

        for percent\_complete in range(percent):

            time.sleep(0.1)

            mybar.progress(percent\_complete+1,text="Target Percentage")

def sideBar():

    with st.sidebar:

        selected=option\_menu(

            menu\_title="Main Menu",

            options=["Home", "Progress"],

            icons=["house", "eye"],

            menu\_icon="cast",

            default\_index=0

        )

    if selected=="Home":

        st.subheader(f"Page: {selected}")

        Home()

        graphs()

    if selected=="Progress":

        st.subheader(f"Page: {selected}")

        Progressbar()

        graphs()

sideBar()

#theme

hide\_st\_style="""

<style>

#MainMenu{visibility:hidden;}

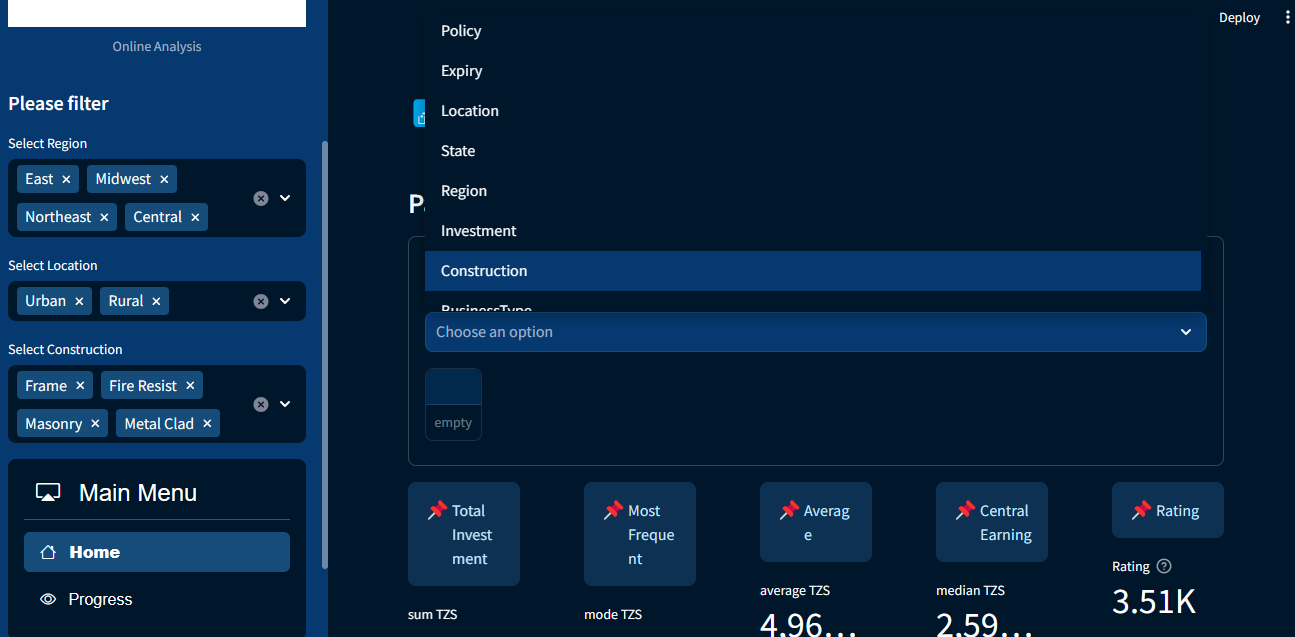
footer{visibility:hidden;}

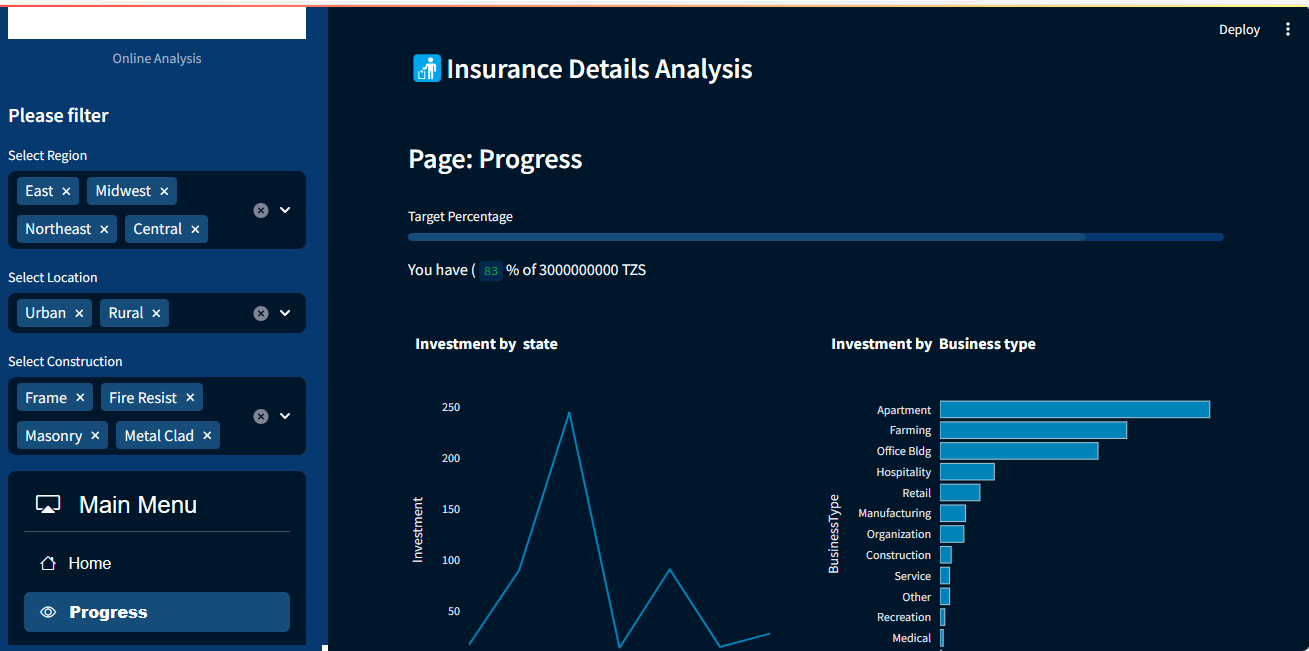
header{visibility:hidden;}

</style>

"""

**Output:**

****

****