import streamlit as st

import pandas as pd

import warnings

import plotly.express as px

import plotly.graph\_objects as go

from datetime import date

import datetime

# Config

warnings.filterwarnings("ignore")

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

# --- Authentication ---

user\_roles = {

"APCMAY": {"division": "Osvita", "email": "venkateshbabu.pr@abbott.com"},

"APCMAY1": {"division": "Endura", "email": "arijit.gupta@abbott.com"},

"APCMAY2": {"division": "General Medicine", "email": "basheer.ahmed@abbott.com"},

"APCMAY3": {"division": "Multi Therapy", "email": "nayan.borthakhur@abbott.com"},

"APCMAY4": {"division": "NovaNXT", "email": "kailash.parihar@abbott.com"}

}

if "authenticated" not in st.session\_state:

st.session\_state.authenticated = False

st.session\_state.username = None

st.session\_state.user\_division = None

if not st.session\_state.authenticated:

st.title("🔐 Login Page")

username = st.text\_input("Enter your username:")

email = st.text\_input("Enter your email:")

if username in user\_roles and user\_roles[username]["email"] == email:

st.session\_state.authenticated = True

st.session\_state.username = username

st.session\_state.user\_division = user\_roles[username]["division"]

st.success(f"Welcome {username}! You are authenticated under {st.session\_state.user\_division} division.")

else:

st.error("Access denied. Contact admin for access.")

st.stop()

# --- Load Data ---

file\_path = r"C:\Users\NADARJX\OneDrive - Abbott\Documents\APC KPI\KPI new- Jun 2025.xlsx"

file\_path1 = r"C:\Users\NADARJX\OneDrive - Abbott\Documents\APC KPI\Chronic Missing Report APC - Mar to May.xlsx"

file\_path2 = r"C:\Users\NADARJX\OneDrive - Abbott\Documents\APC KPI\Comex\_Apc.xlsx"

df = pd.read\_excel(file\_path)

df1 = pd.read\_excel(file\_path1, sheet\_name="Base Data", engine="openpyxl")

df2 = pd.read\_excel(file\_path2)

# --- Page Navigation ---

st.sidebar.title("📁 Dashboard Menu")

page = st.sidebar.radio("Go to", ["📊 KPI Dashboard", "📉 Missed Doctors Report"])

# Common division filtering

division = st.session\_state.user\_division

# --- Page 1: KPI Dashboard ---

if page == "📊 KPI Dashboard":

# All your original page 1 logic (charts, filters, KPIs, col10 bar chart) goes here

# Shortened in this snippet for brevity

st.title("📊 KPI Dashboard")

st.write("All visualizations (Col1–Col10) shown here from your current dashboard code.")

# Paste everything from your current "KPI Dashboard" code here (excluding fig1, fig2)

# --- Page 2: Missed Doctors Report ---

elif page == "📉 Missed Doctors Report":

st.title("📉 Missed Doctors - Last 3 Months")

# Apply RLS for authenticated user

df\_filtered1 = df1[df1["Divison Name"] == division]

# Sidebar Filters

st.sidebar.header("🎯 Filters (Missed Doctors)")

for column, label in [

("Divison Name", "Division"),

("TBM Name", "TBM Name"),

("ABM Name", "ABM"),

("ZBM Name", "ZBM"),

("Month", "Month"),

("To be Met", "Frequency"),

]:

options = df\_filtered1[column].dropna().unique()

selected = st.sidebar.multiselect(f"Select {label}", options)

if selected:

df\_filtered1 = df\_filtered1[df\_filtered1[column].isin(selected)]

# Chart 1: Unique Doctor Count by Specialty

st.markdown("## \*\*Unique Doctors Missed in Division (Last 3 Months)\*\*")

specialty\_counts = df\_filtered1.groupby('Specialty By Practice')['Customer Code'].nunique().reset\_index()

specialty\_counts = specialty\_counts.sort\_values(by='Customer Code', ascending=False)

fig1 = px.bar(

specialty\_counts,

x='Specialty By Practice',

y='Customer Code',

text='Customer Code',

labels={'Customer Code': 'Unique Customer Count'},

color\_discrete\_sequence=["#E6ADDE"]

)

fig1.update\_traces(texttemplate="<b>%{y:.0f}</b>", textfont=dict(size=16, color="black"), width=0.8)

st.plotly\_chart(fig1, use\_container\_width=True)

# Chart 2: Frequency by Specialty

st.markdown("### \*\*Total Frequency of Doctors 1, 2, 3\*\*")

division\_names = df\_filtered1['Divison Name'].unique()

selected\_division\_chart = st.selectbox("Select Division Name", division\_names)

filtered\_data = df\_filtered1[df\_filtered1['Divison Name'] == selected\_division\_chart]

frequency\_data = (

filtered\_data.groupby('Specialty By Practice')['To be Met']

.sum().reset\_index().sort\_values(by='To be Met', ascending=False)

)

fig2 = px.bar(

frequency\_data,

x='Specialty By Practice',

y='To be Met',

text='To be Met',

labels={'To be Met': 'Frequency', 'Specialty By Practice': 'Specialty'},

color\_discrete\_sequence=["#008004"]

)

fig2.update\_traces(texttemplate="<b>%{y:.0f}</b>", textfont=dict(size=16, color="black"), width=0.8)

st.plotly\_chart(fig2, use\_container\_width=True)

# Summary Table

st.subheader("📋 Missing HCP Details Summary")

st.dataframe(df\_filtered1[['Customer Code', 'HCP Name', 'Specialty By Practice', 'To be Met']])