import streamlit as st

import pandas as pd

from datetime import datetime

import io

st.set\_page\_config(page\_title="Occasion Pricing Advisor", page\_icon="💃", layout="centered")

st.title("💃 Occasion Pricing Advisor")

st.caption("Upload an item, enter its original price, and get suggested rental prices by season/occasion.")

# ---- User Profile (stored in session for demo) ----

st.sidebar.header("Your Profile")

user\_type = st.sidebar.radio("Are you in high school or college?", ["highschool", "college"])

region = st.sidebar.text\_input("Region (optional)", value="Dallas, TX")

st.sidebar.markdown("---")

st.sidebar.caption("Tip: you can tweak occasion multipliers in `occasion\_calendar.csv`.")

# ---- Occasion Calendar ----

@st.cache\_data

def load\_calendar():

return pd.read\_csv("occasion\_calendar.csv")

calendar = load\_calendar()

# ---- Item Upload Form ----

st.subheader("1) Item details")

img = st.file\_uploader("Upload a photo (JPG/PNG)", type=["jpg","jpeg","png"])

col1, col2 = st.columns(2)

with col1:

original\_price = st.number\_input("Original retail price ($)", min\_value=10.0, max\_value=5000.0, value=250.0, step=5.0)

condition = st.slider("Condition (1-poor to 5-excellent)", min\_value=1, max\_value=5, value=5)

material = st.selectbox("Material (optional)", ["Unknown","Silk","Satin","Cotton","Lace","Polyester","Sequin","Other"])

with col2:

silhouette = st.selectbox("Silhouette (optional)", ["Unknown","mini","midi","gown","set","jumpsuit"])

color = st.selectbox("Color (optional)", ["Unknown","black","white","pink","blue","red","green","gold","silver","other"])

notes = st.text\_area("Notes (damage, fit, brand, etc.)")

st.subheader("2) Pricing knobs (optional)")

base\_pct = st.slider("Base rental % of retail", 5, 50, 30, help="Starting point before season/condition adjustments")

rush\_markup = st.slider("Rush (<4 days to event) markup %", 0, 50, 10)

weekend\_markup = st.slider("Weekend event markup %", 0, 40, 5)

st.subheader("3) Event info (for rush/weekend logic)")

event\_date = st.date\_input("Next expected event date (optional)")

today = datetime.now().date()

days\_to\_event = (event\_date - today).days if event\_date else None

is\_weekend = event\_date.weekday() >= 5 if event\_date else False

def material\_adjust(material):

return {

"Silk": 1.10, "Satin": 1.07, "Lace": 1.05, "Sequin": 1.12,

"Cotton": 1.00, "Polyester": 0.98, "Other": 1.00, "Unknown": 1.00,

}.get(material, 1.00)

def condition\_adjust(score):

# scale 1..5 -> 0.75 .. 1.05

return {1:0.75,2:0.85,3:0.93,4:1.00,5:1.05}[int(score)]

def silhouette\_adjust(sil):

return {"mini":1.00,"midi":1.05,"gown":1.15,"set":1.02,"jumpsuit":0.95,"Unknown":1.00}.get(sil,1.00)

def rush\_weekend\_multiplier(days, weekend, rush\_pct, weekend\_pct):

m = 1.0

if days is not None and days <= 4:

m \*= (1 + rush\_pct/100.0)

if weekend:

m \*= (1 + weekend\_pct/100.0)

return m

if st.button("Generate Pricing Report"):

# Base rental before occasion

base\_price = original\_price \* (base\_pct/100.0)

base\_price \*= material\_adjust(material)

base\_price \*= condition\_adjust(condition)

base\_price \*= silhouette\_adjust(silhouette)

base\_price \*= rush\_weekend\_multiplier(days\_to\_event, is\_weekend, rush\_markup, weekend\_markup)

# Build occasion table filtered by user\_type

df = calendar[calendar["user\_type"]==user\_type].copy()

# Compute seasonality factor by today's month as a demo and also provide a general suggested price

month = today.month

df["in\_season\_now"] = ( (df["start\_month"] <= df["end\_month"]) & ( (month >= df["start\_month"]) & (month <= df["end\_month"]) ) ) | \

( (df["start\_month"] > df["end\_month"]) & ( (month >= df["start\_month"]) | (month <= df["end\_month"]) ) )

df["suggested\_price"] = (base\_price \* df["multiplier"]).round(0)

df["low"] = (df["suggested\_price"] \* 0.9).round(0)

df["high"] = (df["suggested\_price"] \* 1.1).round(0)

# Simple confidence score

conf = 70

if material in ["Silk","Sequin"] and condition >=4: conf += 5

if silhouette=="gown": conf += 5

df["confidence\_%"] = conf

st.image(img, caption="Uploaded item", use\_column\_width=True) if img else None

st.markdown(f"\*\*Base rental (pre-season):\*\* ${base\_price:.0f}")

st.dataframe(df[["occasion","start\_month","end\_month","multiplier","suggested\_price","low","high","in\_season\_now","confidence\_%"]])

# Downloadable CSV

st.download\_button("Download report CSV", data=df.to\_csv(index=False), file\_name="pricing\_report.csv", mime="text/csv")

st.markdown("""

\*\*How this was computed:\*\*

Base = retail × base% × material × condition × silhouette × (rush/weekend) → then × occasion multiplier.

You can edit multipliers in `occasion\_calendar.csv`.

""")