requirements.txt

streamlit

import math

import streamlit as st

def calculate\_custom\_price(garment\_type, color, quantity, decoration\_method, placement='left chest', stitch\_count=0, extra\_ink\_cc=0):

prices = {

'Comfort Colors Pocket': {'light': (24, 28), 'dark': (26, 30)},

'Comfort Colors 1717': {'light': (24, 28), 'dark': (25, 29)},

'Port & Co Pocket (PC54P)': {'light': (22, 25), 'dark': (23, 26)},

'Port & Co Non-Pocket (PC54)': {'light': (21, 24), 'dark': (22, 25)},

'Tultex 202': {'light': (19, 22), 'dark': (20, 23)},

'Gildan': {'light': (16, 22), 'dark': (16, 22)},

'Nike Polo (NKDC1963)': {'any': (59.95, 64.95)},

'Adidas Polo (A230)': {'any': (59.95, 64.95)},

'Port & Co Polo (K110)': {'any': (36.95, 40)},

'SportTek Polo (ST550)': {'any': (34.95, 36.95)},

'Sportsman Hat (AH35)': {'any': (23, 25)},

}

embroidery\_pricing = [

(2000, [7.75, 3.91, 3.45, 3.06, 2.88, 2.41]),

(4000, [9.08, 5.11, 4.37, 3.45, 3.22, 2.76]),

(6000, [9.95, 5.87, 4.95, 4.03, 3.57, 3.22]),

(8000, [10.29, 6.32, 5.29, 4.37, 4.03, 3.57]),

(10000, [10.93, 6.90, 5.87, 4.89, 4.60, 4.03]),

(12000, [11.50, 7.89, 6.32, 5.40, 4.95, 4.30]),

]

placement\_fees = {

'light': {

'left chest': [2.75, 2.40, 2.05, 1.70],

'full front/back': [4.25, 4.00, 2.75, 3.50]

},

'dark': {

'left chest': [3.75, 3.40, 3.05, 2.70],

'full front/back': [6.00, 5.65, 5.30, 4.95]

}

}

if garment\_type not in prices:

raise ValueError("Unknown garment type")

color\_key = 'any' if 'any' in prices[garment\_type] else ('dark' if color.lower() == 'dark' else 'light')

price\_range = prices[garment\_type][color\_key]

if quantity < 24:

base\_price = price\_range[1]

elif quantity < 51:

base\_price = (price\_range[0] + price\_range[1]) / 2

elif quantity < 145:

base\_price = price\_range[0] \* 0.95

else:

base\_price = price\_range[0] \* 0.90

final\_price = base\_price

if decoration\_method.lower() == 'embroidery':

stitch\_bracket = next((pricing for limit, pricing in embroidery\_pricing if stitch\_count <= limit), embroidery\_pricing[-1][1])

if quantity < 6:

embroidery\_cost = stitch\_bracket[0]

elif quantity < 25:

embroidery\_cost = stitch\_bracket[1]

elif quantity < 73:

embroidery\_cost = stitch\_bracket[2]

elif quantity < 145:

embroidery\_cost = stitch\_bracket[3]

elif quantity < 289:

embroidery\_cost = stitch\_bracket[4]

else:

embroidery\_cost = stitch\_bracket[5]

final\_price += embroidery\_cost

elif decoration\_method.lower() in ['screenprint', 'dtg']:

placement\_options = placement\_fees[color\_key]

if placement.lower() not in placement\_options:

raise ValueError("Unknown placement option")

if quantity < 25:

placement\_cost = placement\_options[placement.lower()][0]

elif quantity < 51:

placement\_cost = placement\_options[placement.lower()][1]

elif quantity < 145:

placement\_cost = placement\_options[placement.lower()][2]

else:

placement\_cost = placement\_options[placement.lower()][3]

ink\_surcharge = max(0, extra\_ink\_cc - 7) \* 1.0 if color\_key == 'dark' else 0

final\_price += placement\_cost + ink\_surcharge

return round(final\_price, 2)

# Streamlit web app

st.title("Campus Classics Quick Quote Calculator")

garment = st.selectbox('Select Garment Type', list(prices.keys()))

color = st.selectbox('Select Garment Color', ['light', 'dark'])

quantity = st.number\_input('Enter Quantity', min\_value=1, value=24)

decoration = st.selectbox('Decoration Method', ['screenprint', 'dtg', 'embroidery'])

placement = st.selectbox('Placement (for screenprint/DTG)', ['left chest', 'full front/back'])

stitch\_count = st.number\_input('Stitch Count (for embroidery)', min\_value=0, value=0)

extra\_ink\_cc = st.number\_input('Extra Ink CCs over 7 (for DTG)', min\_value=0, value=0)

if st.button('Calculate Quote'):

quote = calculate\_custom\_price(garment, color, quantity, decoration, placement, stitch\_count, extra\_ink\_cc)

st.success(f"Quick quote: ${quote} per item")