# generate\_enhanced\_sales\_ppt.py

# This script generates an enhanced PowerPoint presentation (.pptx) with colors, charts, and graphs.

# It includes bar charts, pie charts, and line graphs using matplotlib-generated images.

# To run: Save this as generate\_enhanced\_sales\_ppt.py, install required packages with:

# 'pip install python-pptx matplotlib', then 'python generate\_enhanced\_sales\_ppt.py'

# Output: Enhanced\_Sales\_Performance\_Report.pptx

from pptx import Presentation

from pptx.util import Inches, Pt

from pptx.enum.text import PP\_ALIGN, MSO\_THEME\_COLOR\_INDEX

from pptx.dml.color import RGBColor

from pptx.enum.dml import MSO\_THEME\_COLOR

import matplotlib.pyplot as plt

import io

import base64

# Create a new presentation

prs = Presentation()

# Generate charts as base64 images (using sample data from notes)

def generate\_charts():

charts = {}

# Chart 1: 2015 Agent Revenue Bar Chart

fig1, ax1 = plt.subplots(figsize=(6, 4))

agents\_2015 = ['Tolu', 'Tonye']

revenues\_2015 = [2883445, 289.12]

bars1 = ax1.bar(agents\_2015, revenues\_2015, color=['#2E8B57', '#DC143C'])

ax1.set\_title('2015 Agent Revenue', fontsize=14, color='blue')

ax1.set\_ylabel('Revenue', color='green')

ax1.bar\_label(bars1, fmt='%.0f')

plt.xticks(rotation=45)

plt.tight\_layout()

buf1 = io.BytesIO()

fig1.savefig(buf1, format='png', bbox\_inches='tight')

buf1.seek(0)

charts['agent\_2015'] = base64.b64encode(buf1.read()).decode('utf-8')

plt.close(fig1)

# Chart 2: Product Performance Over 2 Years (Subplots: Units and Revenue)

fig2, (ax21, ax22) = plt.subplots(1, 2, figsize=(10, 4))

products = ['HP', 'Apple']

units\_sold = [722, 10]

revenues = [955000, 1500]

bars2a = ax21.bar(products, units\_sold, color=['#4169E1', '#FF8C00'])

ax21.set\_title('Units Sold', fontsize=12, color='navy')

ax21.bar\_label(bars2a, fmt='%d')

bars2b = ax22.bar(products, revenues, color=['#4169E1', '#FF8C00'])

ax22.set\_title('Revenue', fontsize=12, color='navy')

ax22.bar\_label(bars2b, fmt='%.0f')

plt.tight\_layout()

buf2 = io.BytesIO()

fig2.savefig(buf2, format='png', bbox\_inches='tight')

buf2.seek(0)

charts['product\_overall'] = base64.b64encode(buf2.read()).decode('utf-8')

plt.close(fig2)

# Chart 3: 2015 Branch Performance Pie Chart

fig3, ax3 = plt.subplots(figsize=(6, 4))

branches = ['Ijoh', 'GRT']

percentages = [70.45, 7.8]

ax3.pie(percentages, labels=branches, autopct='%1.1f%%', colors=['#87CEEB', '#F08080'], startangle=90)

ax3.set\_title('2015 Branch Performance %', fontsize=14, color='purple')

buf3 = io.BytesIO()

fig3.savefig(buf3, format='png', bbox\_inches='tight')

buf3.seek(0)

charts['branch\_2015'] = base64.b64encode(buf3.read()).decode('utf-8')

plt.close(fig3)

# Chart 4: Revenue by Month Line Chart

fig4, ax4 = plt.subplots(figsize=(6, 4))

months = ['March', 'July', 'December']

rev = [167.44, 1646, 2000] # Assuming higher for Dec based on notes

ax4.plot(months, rev, marker='o', color='#8A2BE2', linewidth=2, markersize=8)

ax4.set\_title('Revenue by Month Over 2 Years', fontsize=14, color='darkgreen')

ax4.set\_ylabel('Revenue', color='darkgreen')

ax4.grid(True, linestyle='--', alpha=0.7)

plt.xticks(rotation=45)

plt.tight\_layout()

buf4 = io.BytesIO()

fig4.savefig(buf4, format='png', bbox\_inches='tight')

buf4.seek(0)

charts['monthly'] = base64.b64encode(buf4.read()).decode('utf-8')

plt.close(fig4)

return charts

charts = generate\_charts()

# Helper function to add a slide with title, bullets, and optional chart

def add\_slide(prs, title, bullets=None, chart\_key=None, layout=1):

slide\_layout = prs.slide\_layouts[layout]

slide = prs.slides.add\_slide(slide\_layout)

title\_shape = slide.shapes.title

title\_shape.text = title

title\_shape.text\_frame.paragraphs[0].runs[0].font.color.rgb = RGBColor(0, 51, 102) # Navy blue

title\_shape.text\_frame.paragraphs[0].runs[0].font.size = Pt(24)

title\_shape.text\_frame.paragraphs[0].alignment = PP\_ALIGN.CENTER

if bullets:

body\_shape = slide.placeholders[1]

tf = body\_shape.text\_frame

tf.clear()

for bullet in bullets:

p = tf.add\_paragraph()

p.text = bullet

p.level = 1 if bullet.startswith(' ') else 0

p.alignment = PP\_ALIGN.LEFT

for run in p.runs:

run.font.size = Pt(16)

run.font.color.rgb = RGBColor(0, 0, 0) # Black text

# Add chart if provided

if chart\_key and chart\_key in charts:

img\_data = charts[chart\_key]

img\_stream = io.BytesIO(base64.b64decode(img\_data))

left = Inches(1)

top = Inches(2)

pic = slide.shapes.add\_picture(img\_stream, left, top, width=Inches(6))

# Add colored footer

footer\_left = Inches(0.5)

footer\_top = Inches(6.5)

footer\_txbox = slide.shapes.add\_textbox(footer\_left, footer\_top, Inches(9), Inches(0.5))

footer\_frame = footer\_txbox.text\_frame

footer\_frame.text = "United to Prevent Violence Against Women and Girls | www.cewhin.com"

footer\_frame.paragraphs[0].alignment = PP\_ALIGN.CENTER

for run in footer\_frame.paragraphs[0].runs:

run.font.size = Pt(12)

run.font.color.rgb = RGBColor(139, 69, 19) # Saddle brown

return slide

# Slide 1: Title Slide (with color)

title\_slide\_layout = prs.slide\_layouts[0]

slide = prs.slides.add\_slide(title\_slide\_layout)

title = slide.shapes.title

title.text = "Sales Performance Report: 2015-2016"

title.text\_frame.paragraphs[0].runs[0].font.color.rgb = RGBColor(0, 128, 0) # Green

title.text\_frame.paragraphs[0].runs[0].font.size = Pt(32)

subtitle = slide.placeholders[1]

subtitle.text = "Key Insights from Branch Data"

subtitle.text\_frame.paragraphs[0].runs[0].font.color.rgb = RGBColor(100, 100, 100) # Gray

# Footer

footer\_txbox = slide.shapes.add\_textbox(Inches(0.5), Inches(6.5), Inches(9), Inches(0.5))

footer\_frame = footer\_txbox.text\_frame

footer\_frame.text = "United to Prevent Violence Against Women and Girls | www.cewhin.com"

footer\_frame.paragraphs[0].alignment = PP\_ALIGN.CENTER

for run in footer\_frame.paragraphs[0].runs:

run.font.size = Pt(12)

run.font.color.rgb = RGBColor(139, 69, 19)

# Slide 2: 2015 Agent Performance + Chart

add\_slide(prs, "2015 Agent Performance", [

"Tolu had the highest sales revenue",

" Total Rev: 2,883,445 / 21.08%",

"Tonye had the lowest sales revenue",

" Total Per: 289.12 / 2.79%"

], 'agent\_2015')

# Slide 3: 2015 Product Performance

add\_slide(prs, "2015 Product Performance", [

"HP had the highest sales revenue",

" Total Par: 5,814 / 414 units",

"Lenovo had the lowest sales revenue",

" Total Par: 208.80 / 160 units",

"Average Price of the year 2015: 200.00"

])

# Slide 4: 2015 Branch Performance + Pie Chart

add\_slide(prs, "2015 Branch Performance", [

"Ijoh is the highest performing branch",

" Total Rev: 7,305.56 / 70.45%",

"GRT is the lowest performing branch",

" Total Revenue: 808.38 / 7.8%"

], 'branch\_2015')

# Slide 5: 2016 Agent Performance

add\_slide(prs, "2016 Agent Performance", [

"Chinma had the highest sales revenue",

" Total Rev: 3,102 / 33.51%",

"Torbari had the lowest sales revenue",

" Total Rev: 57.71 / 0.62%"

])

# Slide 6: 2016 Product Performance

add\_slide(prs, "2016 Product Performance", [

"Apple had the highest performing sales",

" Total Per: 3,247 / 308 units",

"Apple had the lowest (note: possible data overlap; lowest units: 250 / 2 units)",

"Average price of 2016: 125.00"

])

# Slide 7: 2016 Branch Performance

add\_slide(prs, "2016 Branch Performance", [

"GRA had the highest performance",

" Total Rev: 5,193 / 56%",

"Addah Town had the lowest performance",

" Total Rev: 231.12 / 2.5%"

])

# Slide 8: 2016 Revenue by Month + Line Chart

add\_slide(prs, "2016 Revenue by Month", [

"July had the highest performance",

" Total Rev: 1,646",

"March had the lowest performance",

" Total Rev: 167.44"

], 'monthly')

# Slide 9: Revenue by Branch (Overall)

add\_slide(prs, "Revenue by Branch (Overall)", [

"Ijoh Branch has the highest sales revenue",

" Total Revenue: 11,139.07 / 56.73%",

"Town Branch has the lowest sales revenue",

" Total Revenue: 2,486.42 / 10.67%"

])

# Slide 10: Noticeable Insights

add\_slide(prs, "Noticeable Insights", [

"Only 3 out of 11 agents sold Apple products (Tolu, Emeka, Chinma)",

"Only 5 out of 11 agents sold company products (Blessing, Ibrahim, Torbari, Chinma, Uche)",

"Only 5 out of 11 agents sold HP products – DELL (Emeka, China, George, Blessing, Tolu)",

"All 11 agents sold HP products",

"Only 8 agents out of 11 sold Lenovo products (Tolu, George, Blessing, Tonye, Uche, Chinedu, Ibrahim, Tunde)",

"No agent sold all 5 products",

"Blessing and Uche sold the highest number of distinct products (4)",

"Torbari and Tunde sold the lowest number of distinct products (2)"

])

# Slide 11: Over the 2 Years – Agent & Product Performance + Chart

add\_slide(prs, "Over the 2 Years – Agent & Product Performance", [

"Agent Performance:",

" Emeka had the highest sales revenue: 3,109.44 (15.84%)",

" Torbari had the lowest sales revenue: 536.75 (2.73%)",

"Product Performance:",

" HP has the highest sales revenue: 955k / 722 units sold",

" Apple has the lowest sales revenue: 1.5k / 10 units sold"

], 'product\_overall')

# Slide 12: Over the 2 Years – Year Performance

add\_slide(prs, "Over the 2 Years – Year Performance", [

"2015 is the highest performing year",

" Total Revenue: 10,369.54",

" Units Sold: 943",

"2016 is the lowest performing year",

" Total Revenue: 9,258.39",

" Units Sold: MK (data unclear; possibly missing)"

])

# Slide 13: Revenue by Month Over the 2 Years

add\_slide(prs, "Revenue by Month Over the 2 Years", [

"December has the highest revenue by month",

" We have more sales in December than any other month",

"March has the least revenue by month",

" We have less sales in the month of March"

])

# Slide 14: Thank You / Q&A

add\_slide(prs, "Thank You / Q&A", [

"Summary: Key trends show strong performance in 2015, HP dominance, and seasonal peaks in December/July.",

"Contact: For more details, visit www.cewhin.com"

])

# Save the presentation

prs.save('Enhanced\_Sales\_Performance\_Report.pptx')

print("Enhanced PPTX file generated successfully: Enhanced\_Sales\_Performance\_Report.pptx")

print("Features added: Colored text/titles, embedded charts (bar, pie, line, subplots).")