# 📊Real Estate Sales Dataset Visualization By Ali Ahmad🏢

In [22]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load Excel file
file_path = r"C:\Users\Ali Ahmad\Documents\Al Kabir all data\power bi\Al kabir e
df = pd.read_excel(file_path)

# Show first 5 rows
print(df.head())
```

```
         Client Name  Lead ID    Source       Project Category  DP Received  \
0    Syed Adyan Arif      NaN  Personal         Oasis  5 Marla     300000.0
1        M Ayyaz Khan      NaN  Personal         Oasis  3 Marla     200000.0
2           M Yousaf      NaN  New Lead  Safari Villa   GF 281     600000.0
3  Rakhshanda Firdos      NaN  Old Data         Oasis  3 Marla     200000.0
4  Rakhshanda Firdos      NaN  Old Data         Oasis  3 Marla     200000.0

  Voucher Details Online payments                Date Cash Payments  ...  \
0             NaN             NaN  2025-08-06 00:00:00       1950000  ...
1             NaN             NaN  2025-08-06 00:00:00       1275000  ...
2             NaN             NaN  2025-09-06 00:00:00       4000000  ...
3             NaN             NaN  2025-10-06 00:00:00       1275000  ...
4             NaN             NaN  2025-11-06 00:00:00       1275000  ...

  Token Category Token Amount  Online Payments Token Cash Payments  \
0            NaN          NaN                 NaN                 NaN
1            NaN          NaN                 NaN                 NaN
2            NaN          NaN                 NaN                 NaN
3            NaN          NaN                 NaN                 NaN
4            NaN          NaN                 NaN                 NaN

  Token Book Value  Expected DP Date  Token Sales Person  Token Manager  \
0              NaN               NaN                 NaN            NaN
1              NaN               NaN                 NaN            NaN
2              NaN               NaN                 NaN            NaN
3              NaN               NaN                 NaN            NaN
4              NaN               NaN                 NaN            NaN

  Achieved Book Value % Achieved Revenue %
0                  0.20               0.20
1                  0.13               0.13
2                  0.66               0.67
3                  1.34               4.07
4                   NaN                NaN

[5 rows x 31 columns]
```

In [23]:
```
print(df.info)
```

```
<bound method DataFrame.info of            Client Name  Lead ID    Source       P
roject  Category  \
0      Syed Adyan Arif      NaN  Personal         Oasis   5 Marla
1         M Ayyaz Khan      NaN  Personal         Oasis   3 Marla
2             M Yousaf      NaN  New Lead  Safari Villa    GF 281
3    Rakhshanda Firdos      NaN  Old Data         Oasis   3 Marla
4    Rakhshanda Firdos      NaN  Old Data         Oasis   3 Marla
..                 ...      ...       ...           ...       ...
172      Mudasir Aslam      NaN  New Lead  Safari Villa   5-Marla
173      Fareeha Aslam      NaN  New Lead  Safari Villa   5-Marla
174      Mobeen Jawaid      NaN  New Lead    Kings Town  20-Marla
175      Mobeen Jawaid      NaN  New Lead    Kings Town  20-Marla
176       Shafaqat Ali      NaN  New Lead  Safari Villa   5-Marla

     DP Received                Voucher Details Online payments  \
0       300000.0                            NaN             NaN
1       200000.0                            NaN             NaN
2       600000.0                            NaN             NaN
3       200000.0                            NaN             NaN
4       200000.0                            NaN             NaN
..           ...                            ...             ...
172     550000.0   202 First Floor- Facing Park             NaN
173     600000.0  202 Ground Floor- Facing Park             NaN
174    1200000.0              5-Excutive Block             NaN
175    1200000.0              6-Excutive Block             NaN
176     500000.0              269-Second Floor             NaN

                    Date Cash Payments  ... Token Category Token Amount  \
0    2025-08-06 00:00:00       1950000  ...            NaN          NaN
1    2025-08-06 00:00:00       1275000  ...            NaN          NaN
2    2025-09-06 00:00:00       4000000  ...            NaN          NaN
3    2025-10-06 00:00:00       1275000  ...            NaN          NaN
4    2025-11-06 00:00:00       1275000  ...            NaN          NaN
..                   ...           ...  ...            ...          ...
172  2025-09-13 00:00:00           NaN  ...            NaN          NaN
173  2025-09-13 00:00:00           NaN  ...            NaN          NaN
174  2025-09-12 00:00:00           NaN  ...            NaN          NaN
175  2025-09-12 00:00:00           NaN  ...            NaN          NaN
176  2025-09-13 00:00:00           NaN  ...            NaN          NaN

     Online Payments Token Cash Payments Token Book Value  Expected DP Date  \
0                NaN                 NaN              NaN               NaN
1                NaN                 NaN              NaN               NaN
2                NaN                 NaN              NaN               NaN
3                NaN                 NaN              NaN               NaN
4                NaN                 NaN              NaN               NaN
..               ...                 ...              ...               ...
172              NaN                 NaN              NaN               NaN
173              NaN                 NaN              NaN               NaN
174              NaN                 NaN              NaN               NaN
175              NaN                 NaN              NaN               NaN
176              NaN                 NaN              NaN               NaN

     Token Sales Person  Token Manager Achieved Book Value %  \
0                   NaN            NaN                  0.20
1                   NaN            NaN                  0.13
2                   NaN            NaN                  0.66
3                   NaN            NaN                  1.34
4                   NaN            NaN                   NaN
..                  ...            ...                   ...
```

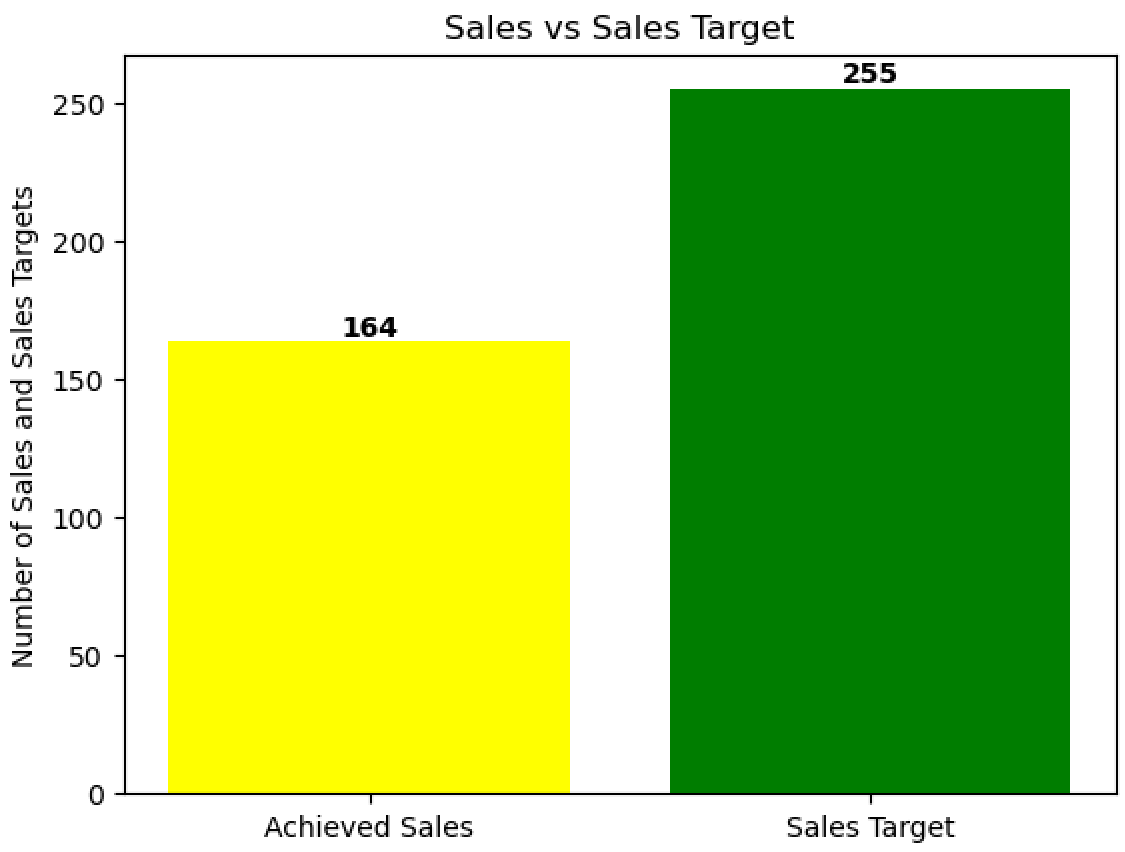
```
172                  NaN            NaN                     NaN
173                  NaN            NaN                     NaN
174                  NaN            NaN                     NaN
175                  NaN            NaN                     NaN
176                  NaN            NaN                     NaN

    Achieved Revenue %
0                 0.20
1                 0.13
2                 0.67
3                 4.07
4                  NaN
..                 ...
172                NaN
173                NaN
174                NaN
175                NaN
176                NaN

[177 rows x 31 columns]>
```

In [24]:
```
print(df.describe())
```

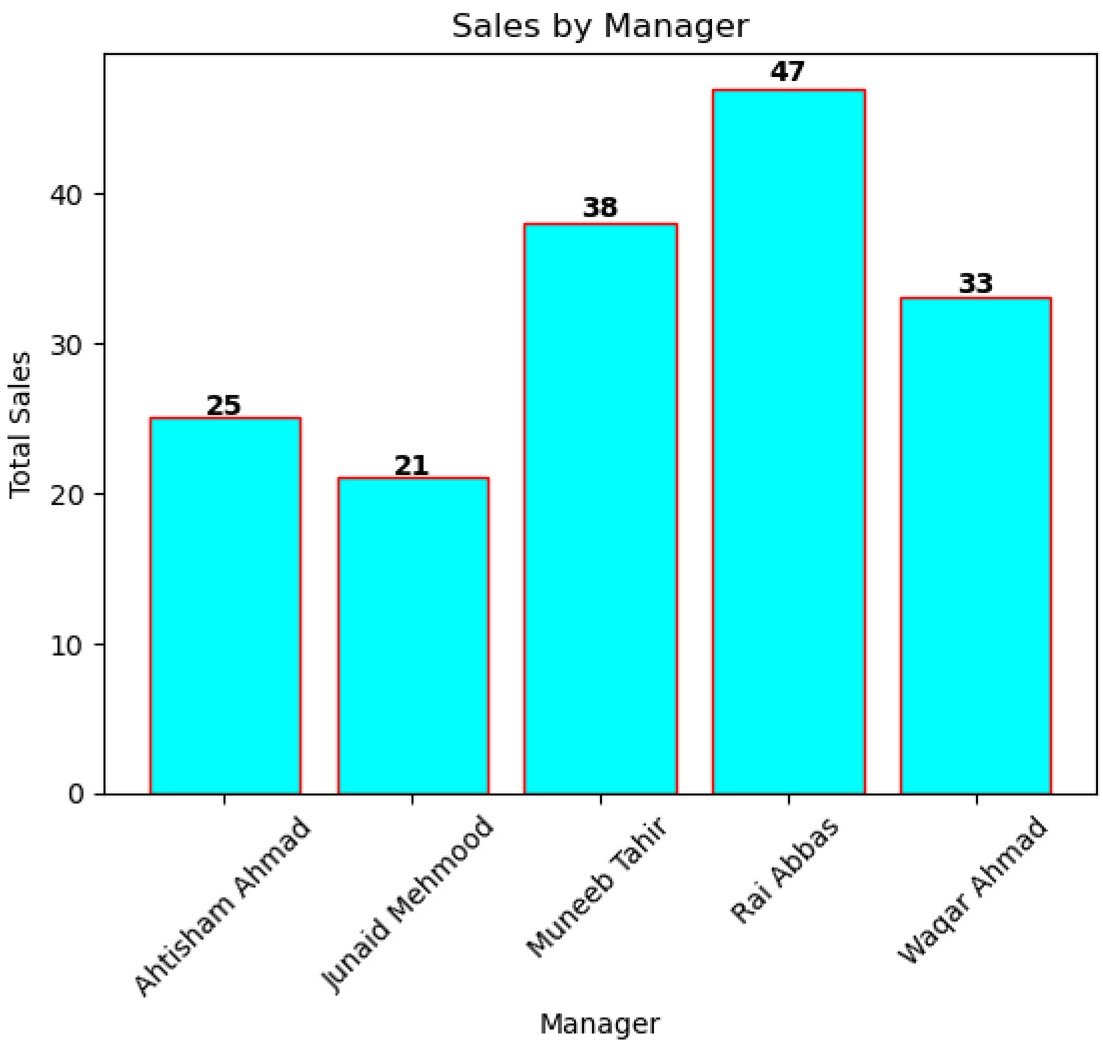
```
       Lead ID   DP Received                    Date_Updated        Sale  \
count      0.0  1.640000e+02                           177  168.000000   
mean       NaN  5.424390e+05  2025-07-30 14:38:38.644067840    0.976190   
min        NaN  2.000000e+05           2025-06-01 00:00:00    0.000000   
25%        NaN  4.000000e+05           2025-07-19 00:00:00    1.000000   
50%        NaN  4.000000e+05           2025-08-03 00:00:00    1.000000   
75%        NaN  5.500000e+05           2025-08-14 00:00:00    1.000000   
max        NaN  2.925000e+06           2025-09-13 00:00:00    1.000000   
std        NaN  3.936596e+05                           NaN    0.152911   

       Sales Target  Book Value Target  Cash Value Traget  Token Amount  \
count     60.000000       6.000000e+01       6.000000e+01     32.000000   
mean       4.250000       8.500000e+06       1.275000e+06   61250.000000   
min        0.000000       0.000000e+00       0.000000e+00   10000.000000   
25%        4.000000       8.000000e+06       1.200000e+06   35000.000000   
50%        5.000000       1.000000e+07       1.500000e+06   50000.000000   
75%        5.000000       1.000000e+07       1.500000e+06  100000.000000   
max        5.000000       1.000000e+07       1.500000e+06  100000.000000   
std        1.373169       2.746338e+06       4.119507e+05   34710.786647   

       Online Payments  Token Cash Payments  Token Book Value  \
count              0.0                  0.0      3.200000e+01   
mean               NaN                  NaN      1.951562e+06   
min                NaN                  NaN      1.425000e+06   
25%                NaN                  NaN      1.500000e+06   
50%                NaN                  NaN      1.500000e+06   
75%                NaN                  NaN      1.500000e+06   
max                NaN                  NaN      4.000000e+06   
std                NaN                  NaN      9.683070e+05   

       Achieved Book Value %  Achieved Revenue %  
count              52.000000           52.000000  
mean                0.639808            1.078077  
min                 0.000000            0.000000  
25%                 0.130000            0.130000  
50%                 0.500000            0.670000  
75%                 0.865000            1.470000  
max                 3.330000            4.910000  
std                 0.654728            1.226267  
```

In [51]:

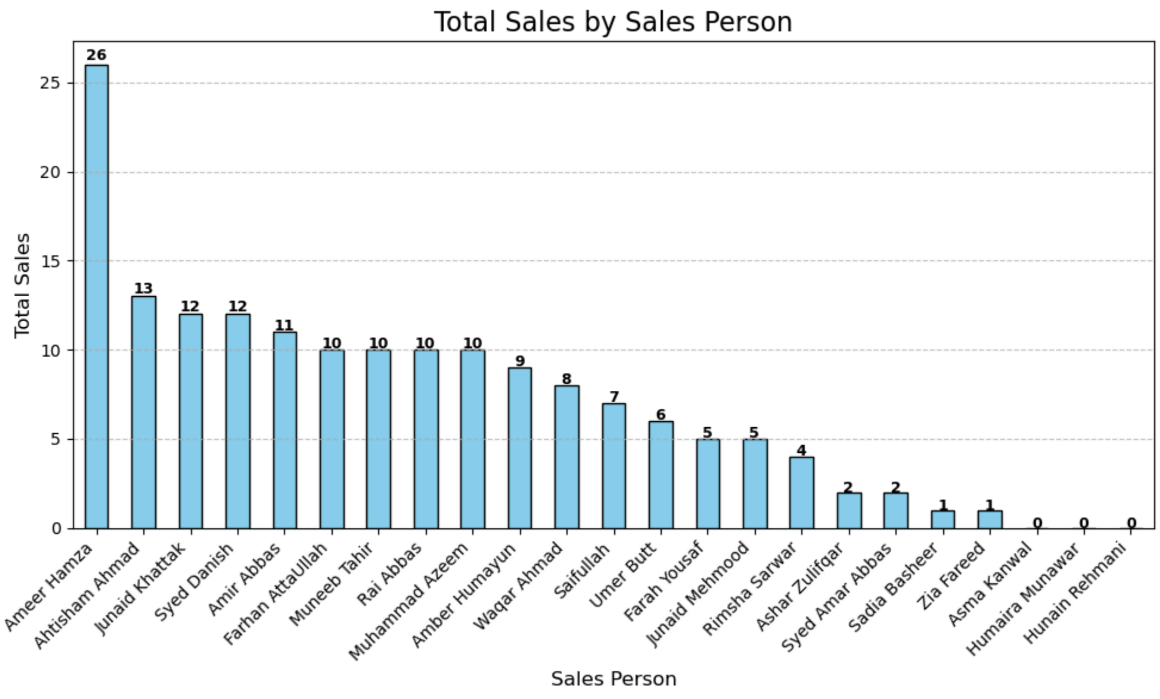
```python
total_sales = df["Sale"].sum()
sales_target = df["Sales Target"].sum()

Values = [total_sales, sales_target]
labels = ["Achieved Sales", "Sales Target"]

plt.bar(labels, Values, color=["Yellow", "green"])

for i, v in enumerate (values):
    plt.text(i, v +(v * 0.01), f"{int(v):,}", ha="center" , fontsize=10, fontwei


plt.title("Sales vs Sales Target")
plt.ylabel("Number of Sales and Sales Targets")

plt.show()
```



In [59]:
```
# Manager-wise total sales
Manager_Sales = df.groupby("Manager")["Sale"].sum().reset_index()

# Plot bar chart
plt.bar(Manager_Sales["Manager"], Manager_Sales["Sale"], color="cyan", edgecolor

# Add value labels on bars
for i, v in enumerate(Manager_Sales["Sale"]):
    plt.text(i, v + (v * 0.01), f"{int(v):,}", ha="center", fontsize=10, fontwei

# Chart formatting
plt.title("Sales by Manager")
plt.xlabel("Manager")
plt.ylabel("Total Sales")
plt.xticks(rotation=45)  # rotate labels if too many managers

# Show chart
plt.show()
```



In [60]:
```
# Group sales by Sales Person
sales_by_person = df.groupby("Sales Person")["Sale"].sum().sort_values(ascending

# Plot bar chart
plt.figure(figsize=(10,6))
bars = sales_by_person.plot(kind="bar", color="skyblue", edgecolor="black")

# Add value labels on top of each bar
for i, v in enumerate(sales_by_person):
    plt.text(i, v + (v * 0.01), f"{int(v):,}", ha="center", fontsize=9, fontweig

# Formatting
plt.title("Total Sales by Sales Person", fontsize=16)
plt.xlabel("Sales Person", fontsize=12)
plt.ylabel("Total Sales", fontsize=12)
plt.xticks(rotation=45, ha="right")
plt.grid(axis="y", linestyle="--", alpha=0.7)

plt.tight_layout()
plt.show()
```



In [70]:
```
# Convert Book Value and Book Value Target to numeric (ignore errors, turn non-n
df["Book Value"] = pd.to_numeric(df["Book Value"], errors="coerce")
df["Book Value Target"] = pd.to_numeric(df["Book Value Target"], errors="coerce"

# --- Calculate KPIs ---
total_sales = df["Sale"].sum()
sales_target = df["Sales Target"].sum()
revenue_achieved = df["Book Value"].sum()
revenue_target = df["Book Value Target"].sum()

# Achievement percentages
sales_achievement = (total_sales / sales_target * 100) if sales_target > 0 else
revenue_achievement = (revenue_achieved / revenue_target * 100) if revenue_targe

# --- Create KPI Dashboard ---
import matplotlib.pyplot as plt
import matplotlib

# ✅ Use emoji-supported font (Windows: Segoe UI Emoji, Mac: Apple Color Emoji,
matplotlib.rcParams['font.family'] = 'Segoe UI Emoji'

plt.figure(figsize=(10,6))
plt.axis("off")  # hide axes

# Add KPI texts
plt.text(0.1, 0.8, f"📊 Total Sales: {total_sales:,.0f}", fontsize=14, fontweigh
plt.text(0.1, 0.7, f"🎯 Sales Target: {sales_target:,.0f}", fontsize=14, fontwe
plt.text(0.1, 0.6, f"💰 Revenue Achieved: {revenue_achieved:,.0f}", fontsize=14,
plt.text(0.1, 0.5, f"🏆 Revenue Target: {revenue_target:,.0f}", fontsize=14, fo
plt.text(0.1, 0.4, f"✅ Sales Achievement: {sales_achievement:.1f}%", fontsize=
plt.text(0.1, 0.3, f"✅ Revenue Achievement: {revenue_achievement:.1f}%", fonts

plt.title("📊 KPI Dashboard", fontsize=18, fontweight="bold")
plt.show()
```

## 📊 KPI Dashboard

📊 Total Sales: 164

🎯 Sales Target: 255

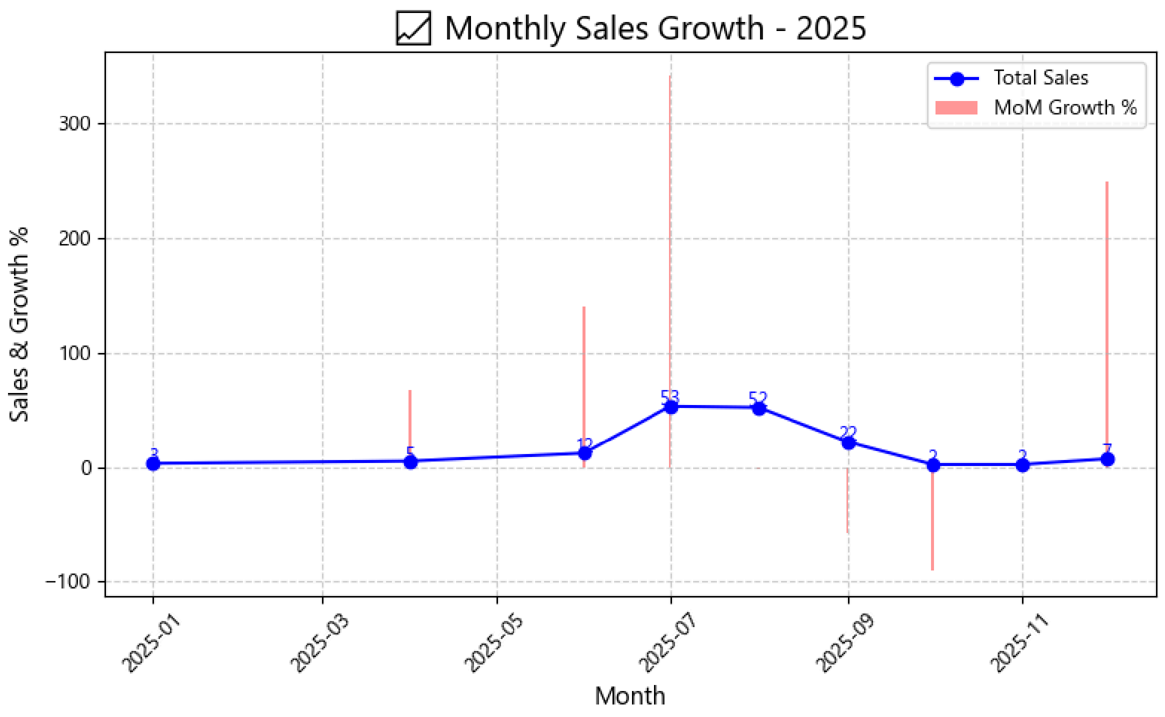
💰 Revenue Achieved: 377,462,250

🏆 Revenue Target: 510,000,000

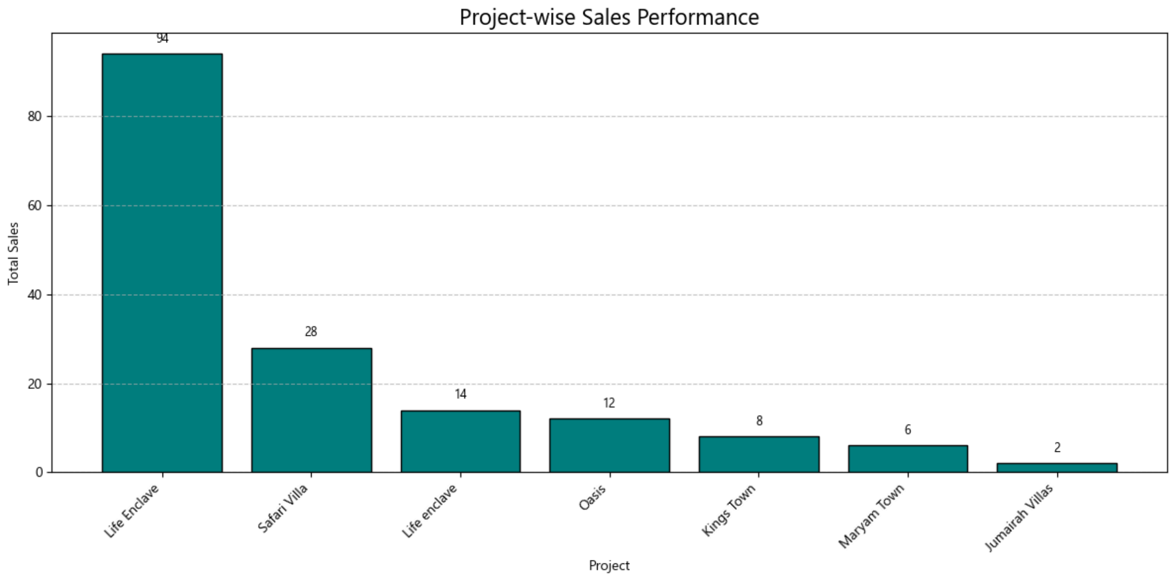
✅ Sales Achievement: 64.3%

☑️ Revenue Achievement: 74.0%

In [76]:
```python
# --- Ensure Date column is datetime ---
df["Date"] = pd.to_datetime(df["Date"], errors="coerce")

# --- Filter only 2025 data ---
df_2025 = df[df["Date"].dt.year == 2025]

# --- Group by Month ---
monthly_sales = df_2025.groupby(df_2025["Date"].dt.to_period("M"))["Sale"].sum()
monthly_sales["Date"] = monthly_sales["Date"].dt.to_timestamp()

# --- Calculate MoM Growth % ---
monthly_sales["Growth %"] = monthly_sales["Sale"].pct_change() * 100

# --- Plot ---
plt.figure(figsize=(8,5))

# Sales Trend
plt.plot(monthly_sales["Date"], monthly_sales["Sale"], marker="o", color="blue",

# Growth %
plt.bar(monthly_sales["Date"], monthly_sales["Growth %"], alpha=0.4, color="red"

# Labels on sales line
for i, v in enumerate(monthly_sales["Sale"]):
    plt.text(monthly_sales["Date"].iloc[i], v + (v*0.02), f"{v:,.0f}", ha="cente

plt.title("📈 Monthly Sales Growth - 2025", fontsize=16, fontweight="bold")
plt.xlabel("Month", fontsize=12)
plt.ylabel("Sales & Growth %", fontsize=12)
plt.xticks(rotation=45)
plt.legend()
plt.grid(linestyle="--", alpha=0.6)

plt.tight_layout()
plt.show()
```



In [79]:
```
import matplotlib.pyplot as plt

# group and sort
project_sales = df.groupby("Project")["Sale"].sum().sort_values(ascending=False)

# plot
fig, ax = plt.subplots(figsize=(12,6))
bars = ax.bar(project_sales.index, project_sales.values, color="teal", edgecolor

# add horizontal value labels above each bar
for bar in bars:
    height = bar.get_height()
    ax.annotate(
        f"{int(height):,}",                          # label text with thousands sep
        xy=(bar.get_x() + bar.get_width() / 2, height),  # point to annotate
        xytext=(0, 6),                               # offset label by 6 points abov
        textcoords="offset points",
        ha="center", va="bottom", fontsize=9, fontweight="bold", rotation=0
    )

# formatting
ax.set_title("Project-wise Sales Performance", fontsize=16, fontweight="bold")
ax.set_xlabel("Project")
ax.set_ylabel("Total Sales")
plt.xticks(rotation=45, ha="right")
ax.grid(axis="y", linestyle="--", alpha=0.7)
plt.tight_layout()
plt.show()

print("🏆 Top 5 Projects by Sales:")
print(project_sales.head(5))
```



```
🏆 Top 5 Projects by Sales:
Project
Life Enclave    94.0
Safari Villa    28.0
Life enclave    14.0
Oasis           12.0
Kings Town       8.0
Name: Sale, dtype: float64
```

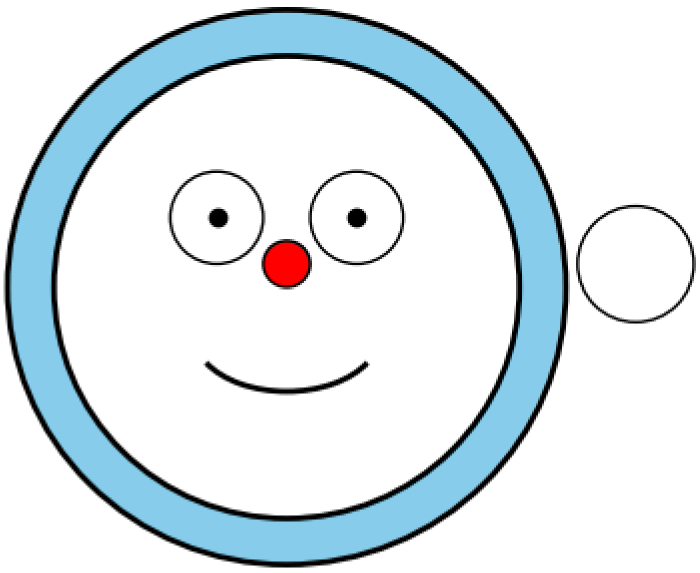
In [84]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

# Example dataframe
# df = pd.DataFrame({"Salesperson": ["Ali", "Ahmed", "Sara", "Usman"],
#                    "Sale": [120000, 95000, 180000, 110000]})

# 1. Find top performer
top_sales = df.groupby("Sales Person")["Sale"].sum().sort_values(ascending=False
top_name = top_sales.index[0]
top_value = top_sales.iloc[0]

# 2. Create a gradient background (for 3D/pro look)
fig, ax = plt.subplots(figsize=(7,7))
x = np.linspace(0, 1, 100).reshape(-1,1)
ax.imshow(x, cmap="coolwarm", interpolation="bicubic", extent=[0,1,0,1], alpha=0

# 3. Add 3D trophy-like emoji
ax.text(0.5, 0.8, "🏆", fontsize=100, ha="center", va="center")

# 4. Add cartoon-like decorations
ax.text(0.2, 0.7, "💰", fontsize=40, alpha=0.8)
ax.text(0.8, 0.7, "📈", fontsize=40, alpha=0.8)
ax.text(0.2, 0.4, "🎯", fontsize=40, alpha=0.8)
ax.text(0.8, 0.4, "🔥", fontsize=40, alpha=0.8)

# 5. Add main text
ax.text(0.5, 0.55, f"Top Performer", fontsize=20, fontweight="bold", color="dark
ax.text(0.5, 0.45, f"{top_name}", fontsize=26, fontweight="bold", color="black",
ax.text(0.5, 0.35, f"Sales: {int(top_value):,}", fontsize=18, color="blue", ha="

# 6. Add animation effect (simulate with sparkles ✨)
for pos in [(0.3,0.9),(0.7,0.9),(0.25,0.2),(0.75,0.2)]:
    ax.text(pos[0], pos[1], "✨", fontsize=20, ha="center", va="center", alpha=

# Formatting
ax.set_xlim(0,1)
ax.set_ylim(0,1)
ax.axis("off")

plt.tight_layout()
plt.show()
```

Top Performer
Ameer Hamza
Sales: 26

In [103…

```
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import matplotlib.animation as animation

# --- Setup figure ---
fig, ax = plt.subplots(figsize=(6,6))
ax.set_xlim(-2, 2)
ax.set_ylim(-2, 2)
ax.axis("off")

# Doraemon-like head
head = patches.Circle((0,0), 1.2, facecolor="skyblue", edgecolor="black", lw=2)
face = patches.Circle((0,0), 1.0, facecolor="white", edgecolor="black", lw=2)
ax.add_patch(head)
ax.add_patch(face)
```

```
# Eyes
eye_left = patches.Circle((-0.3,0.3), 0.2, facecolor="white", edgecolor="black")
eye_right = patches.Circle((0.3,0.3), 0.2, facecolor="white", edgecolor="black")
pupil_left = ax.plot(-0.3,0.3, "o", color="black")[0]
pupil_right = ax.plot(0.3,0.3, "o", color="black")[0]
ax.add_patch(eye_left)
ax.add_patch(eye_right)

# Nose
nose = patches.Circle((0,0.1), 0.1, facecolor="red", edgecolor="black")
ax.add_patch(nose)

# Mouth
mouth = patches.Arc((0,-0.2), 0.8, 0.5, theta1=200, theta2=-20, edgecolor="black
ax.add_patch(mouth)

# Hand (to wave)
hand = patches.Circle((1.5,0.2), 0.25, facecolor="white", edgecolor="black")
ax.add_patch(hand)

# Text with proper emoji font
msg = ax.text(0, -1.6, "See you in next project 👋",
              ha="center", fontsize=14, fontweight="bold", fontname="Segoe UI Em

# --- Animation function ---
def animate(frame):
    # Make the hand wave (move up and down)
    y = 0.2 + 0.1 * ((frame % 20) / 10 - 1)  # wave motion
    hand.set_center((1.5, y))
    return [hand, pupil_left, pupil_right]

# Run animation
ani = animation.FuncAnimation(fig, animate, frames=100, interval=100, blit=True)

plt.show()
```



See you in next project 👋