Project: Tech Usage Tracker

In [9]:

```
#Python libraries used
import pandas as pd             #For data manipulation
import matplotlib.pyplot as plt #For basic plots
import seaborn as sns           #For prettier visualization
```

In [10]:

```
#DataFrame
data = {
    'Date': ['2025-06-20', '2025-06-20', '2025-06-21', '2025-06-21', '2025-06-22'],
    'App': ['Instagram', 'VS Code', 'YouTube', 'Notion', 'WhatsApp'],
    'Category': ['Social', 'Productivity', 'Entertainment', 'Productivity', 'Communication'],
    'Time_spent_min': [60, 90, 50, 45, 30],
    'Purpose': ['Scrolling', 'Coding', 'Relaxing', 'Planning', 'Chatting']
}

tech_usage = pd.DataFrame(data)
tech_usage['Date'] = pd.to_datetime(tech_usage['Date'])  # Convert 'Date' to datetime
tech_usage
```

Out[10]:

	Date	App	Category	Time_spent_min	Purpose
0	2025-06-20	Instagram	Social	60	Scrolling
1	2025-06-20	VS Code	Productivity	90	Coding
2	2025-06-21	YouTube	Entertainment	50	Relaxing
3	2025-06-21	Notion	Productivity	45	Planning
4	2025-06-22	WhatsApp	Communication	30	Chatting

In [14]:

```
#Structure of DataFrame
tech_usage.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Date            5 non-null      datetime64[ns]
 1   App             5 non-null      object
 2   Category        5 non-null      object
 3   Time_spent_min  5 non-null      int64
 4   Purpose         5 non-null      object
dtypes: datetime64[ns](1), int64(1), object(3)
memory usage: 332.0+ bytes
```

In [16]:

```
#Summary Statistics of DataFrame
tech_usage.describe()
```

Out[16]:

	Date	Time_spent_min
count	5	5.00000
mean	2025-06-20 19:12:00	55.00000
min	2025-06-20 00:00:00	30.00000
25%	2025-06-20 00:00:00	45.00000
50%	2025-06-21 00:00:00	50.00000
75%	2025-06-21 00:00:00	60.00000
max	2025-06-22 00:00:00	90.00000
std	NaN	22.36068

In [17]:

```
daily_usage = tech_usage.groupby('Date')['Time_spent_min'].sum().reset_index()
daily_usage
```

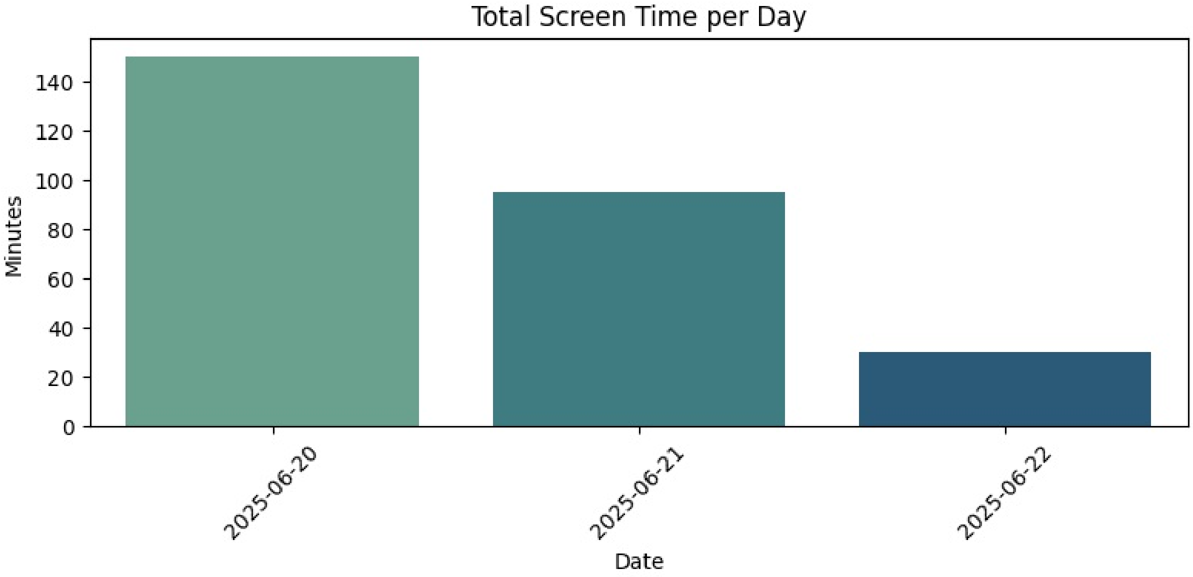
Out[17]:

	Date	Time_spent_min
0	2025-06-20	150
1	2025-06-21	95
2	2025-06-22	30

In [22]:

```
plt.figure(figsize=(8, 4))
```

```
sns.barplot(data=daily_usage, x='Date', y='Time_spent_min', palette='crest')
plt.title('Total Screen Time per Day')
plt.ylabel('Minutes')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



In [24]:

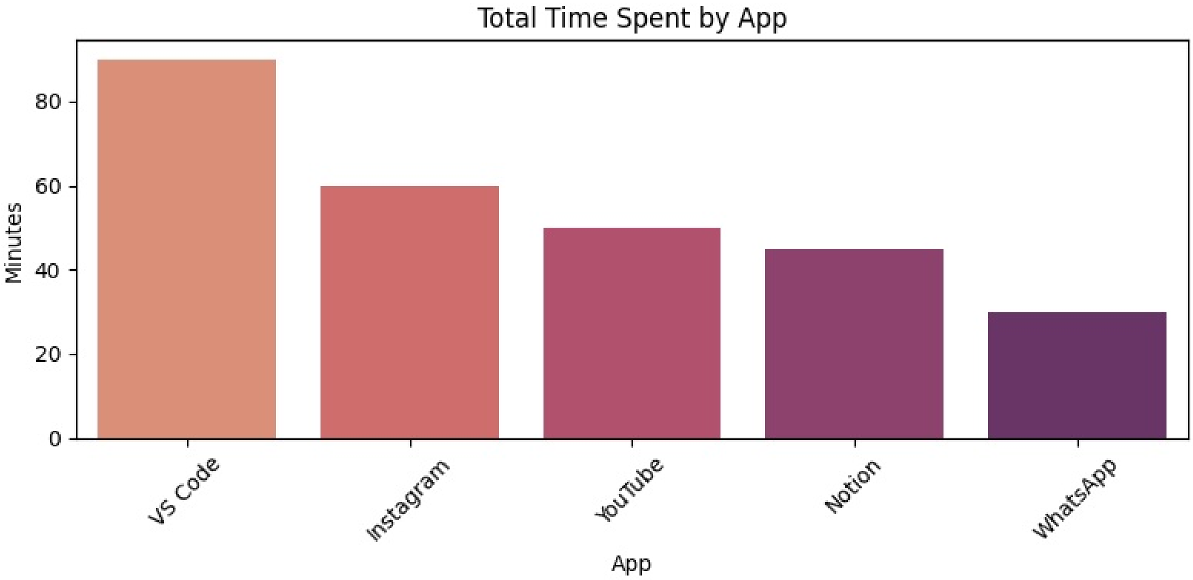
```
app_usage = tech_usage.groupby('App')['Time_spent_min'].sum().reset_index().sort_values(by='Time_spent_min', as
app_usage
```

Out[24]:

	App	Time_spent_min
2	VS Code	90
0	Instagram	60
4	YouTube	50
1	Notion	45
3	WhatsApp	30

In [25]:

```
plt.figure(figsize=(8, 4))
sns.barplot(data=app_usage, x='App', y='Time_spent_min', palette='flare')
plt.title('Total Time Spent by App')
plt.ylabel('Minutes')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



In [26]:

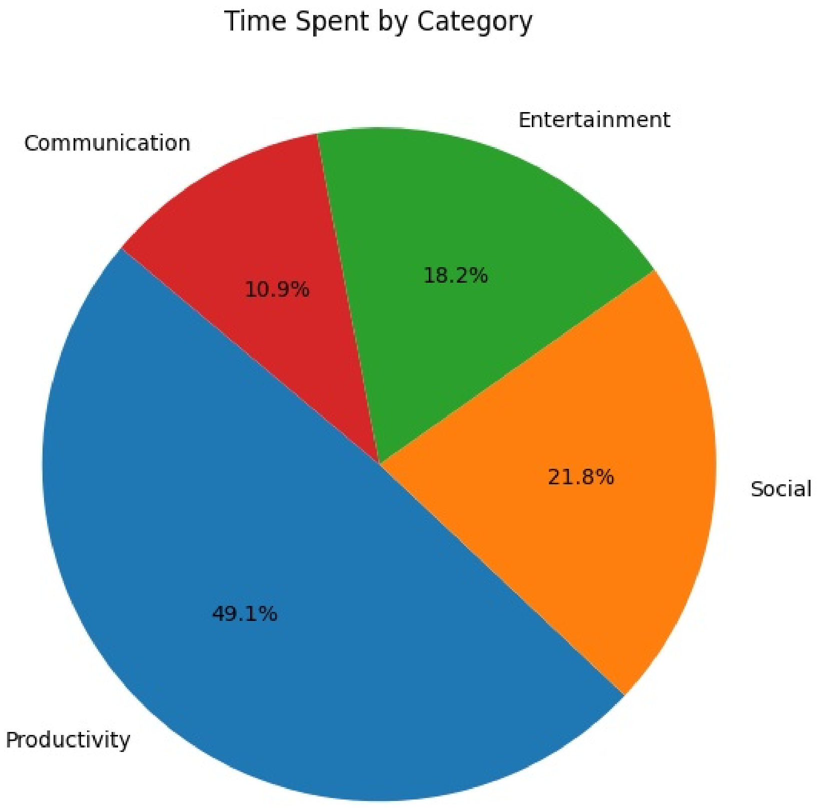
```
category_usage = tech_usage.groupby('Category')['Time_spent_min'].sum().reset_index().sort_values(by='Time_spen
category_usage
```

Out[26]:

	Category	Time_spent_min
2	Productivity	135
3	Social	60
1	Entertainment	50
0	Communication	30

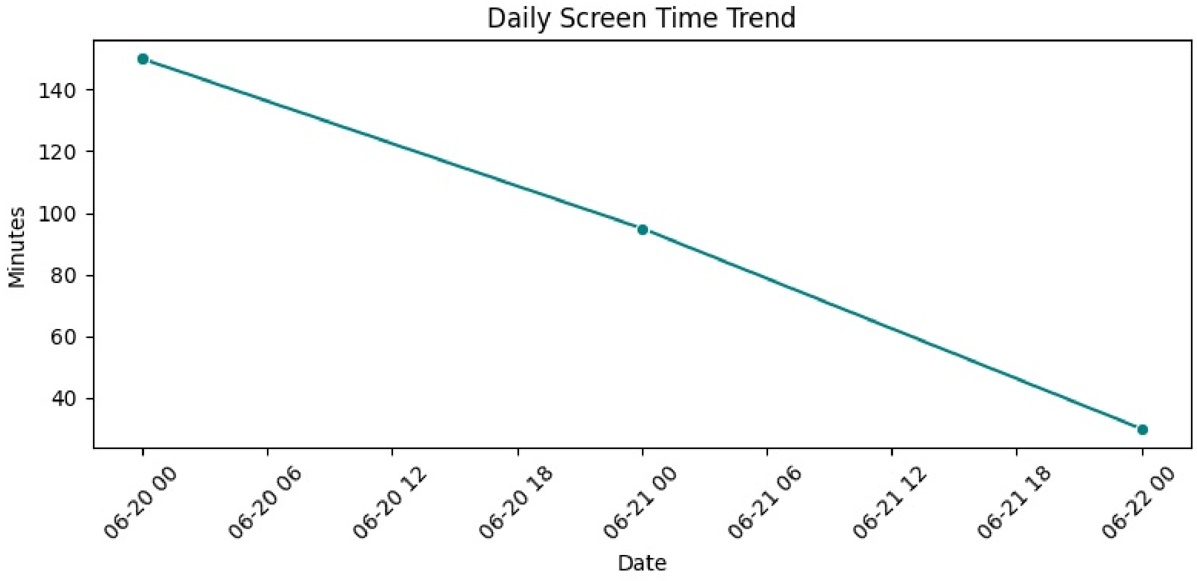
In [27]:

```
plt.figure(figsize=(6, 6))
plt.pie(category_usage['Time_spent_min'], labels=category_usage['Category'], autopct='%1.1f%%', startangle=140)
plt.title('Time Spent by Category')
plt.tight_layout()
plt.show()
```



In [32]:

```
plt.figure(figsize=(8, 4))
sns.lineplot(data=daily_usage, x='Date', y='Time_spent_min', marker='o', color='teal')
plt.title('Daily Screen Time Trend')
plt.ylabel('Minutes')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



In [33]:

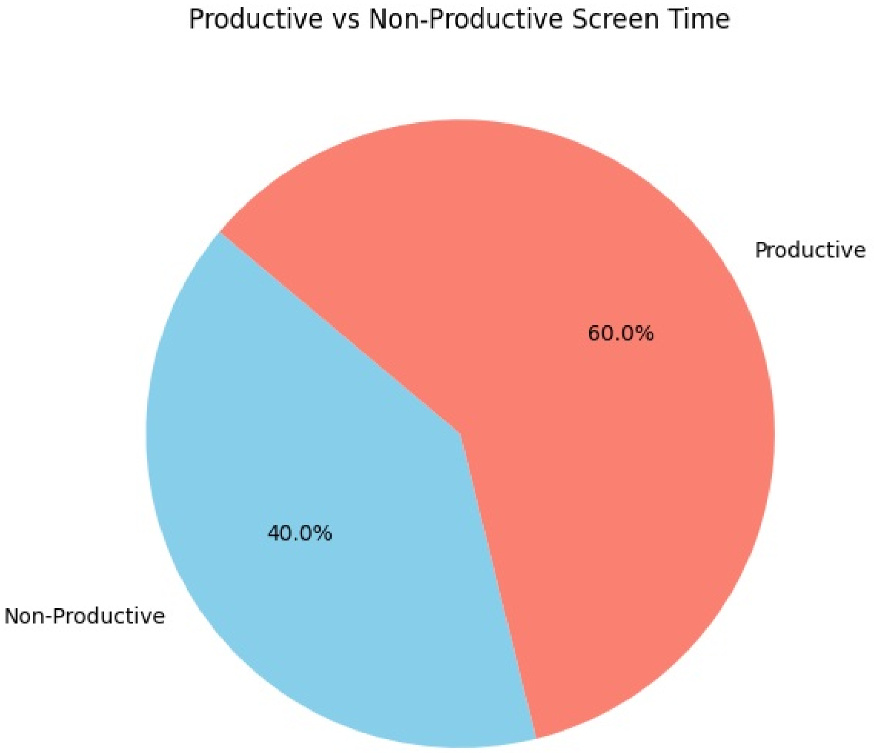
```
# Define which categories are productive
productive_categories = ['Productivity', 'Communication']

# Add a new column to classify each row
tech_usage['Productivity_Type'] = tech_usage['Category'].apply(
    lambda x: 'Productive' if x in productive_categories else 'Non-Productive'
)

# Group by type
productivity_stats = tech_usage.groupby('Productivity_Type')['Time_spent_min'].sum().reset_index()
productivity_stats
```

Out[33]:

	Productivity_Type	Time_spent_min
0	Non-Productive	110
1	Productive	165

In [34]:

```
plt.figure(figsize=(6, 6))
plt.pie(productivity_stats['Time_spent_min'], labels=productivity_stats['Productivity_Type'],
        autopct='%1.1f%%', colors=['skyblue', 'salmon'], startangle=140)
plt.title('Productive vs Non-Productive Screen Time')
plt.tight_layout()
plt.show()
```



In [36]:

```
top_apps = app_usage.head(3)
top_apps
```

Out[36]:

	App	Time_spent_min
2	VS Code	90
0	Instagram	60
4	YouTube	50