CapEx Forecast Model

This notebook demonstrates the full CapEx forecasting flow using:

- `data/sample_capex_input.xlsx`
- `src/helpers.py`

In [27]:
```
import sys
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pathlib import Path

base_dir = Path('..').resolve()
sys.path.append(str(base_dir))

from src.helpers import (
    load_capex_data,
    compute_summaries,
    build_quarterly_capex,
    build_depreciation_schedule,
    build_quarterly_depreciation,
    build_annual_dep_by_program,
    build_project_cashflows,
    export_to_csv,
)

data_path = base_dir / 'data' / 'sample_capex_input.xlsx'
exports_dir = base_dir / 'exports'
exports_dir.mkdir(exist_ok=True)
```

In [28]:
```
# Load dataset
df = load_capex_data(str(data_path))
df.head()
```

Out[28]:

	Asset_ID	Asset_Name	Asset_Type	Process_Area	Scenario	Fab_Location	Quantity
0	TOOL-CVD-3000	CVD 3000 PECVD Chamber	Process Tool	Pioneer CVD	Base	Fab A - Line 1	2
1	TOOL-CVD-3100	CVD 3100 HDP System	Process Tool	Pioneer CVD	Upside	Fab A - Line 2	1
2	TOOL-PVD-2200	PVD 2200 Metallization	Process Tool	Metals	Base	Fab B - Line 3	3
3	TOOL-ETCH-1800	Dry Etch 1800	Process Tool	Etch	Downside	Fab B - Line 1	1
4	TOOL-IMPL-900	Ion Implanter 900	Process Tool	Implant	Base	Fab C - Pilot	1

5 rows × 31 columns

In [29]:
```
df.info()
```

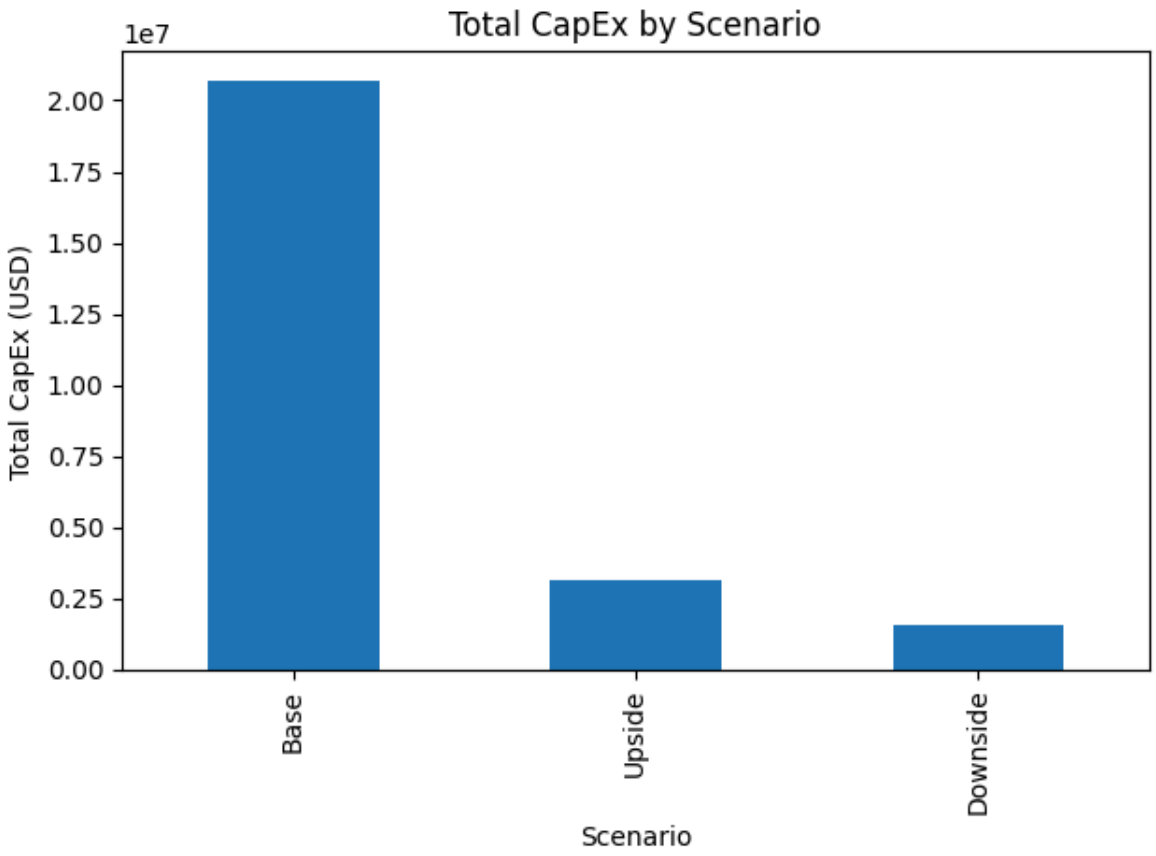
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26 entries, 0 to 25
Data columns (total 31 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Asset_ID                 26 non-null     object
 1   Asset_Name               26 non-null     object
 2   Asset_Type               26 non-null     object
 3   Process_Area             26 non-null     object
 4   Scenario                 26 non-null     object
 5   Fab_Location             26 non-null     object
 6   Quantity                 26 non-null     int64
 7   Unit_Cost_USD            26 non-null     int64
 8   Currency                 26 non-null     object
 9   Lead_Time_Weeks          26 non-null     int64
 10  Order_Quarter            26 non-null     object
 11  Need_Quarter             26 non-null     object
 12  Ramp_Start_Quarter       26 non-null     object
 13  Ramp_Profile             26 non-null     object
 14  Depreciation_Years       26 non-null     int64
 15  Project_Code             26 non-null     object
 16  Program_Name             26 non-null     object
 17  Asset_Class              26 non-null     object
 18  Supplier_Name            26 non-null     object
 19  Region                   24 non-null     object
 20  Criticality              26 non-null     object
 21  Total_Cost_USD           26 non-null     int64
 22  Order_Quarter_Year       26 non-null     int64
 23  Order_Quarter_Num        26 non-null     int64
 24  Need_Quarter_Year        26 non-null     int64
 25  Need_Quarter_Num         26 non-null     int64
 26  Ramp_Start_Quarter_Year  26 non-null     int64
 27  Ramp_Start_Quarter_Num   26 non-null     int64
 28  Order_Q_Index            26 non-null     int64
 29  Ramp_Q_Index             26 non-null     int64
 30  Order_Period             26 non-null     period[Q-DEC]
dtypes: int64(13), object(17), period[Q-DEC](1)
memory usage: 6.4+ KB
```

Portfolio summaries

In [30]:
```
summaries = compute_summaries(df)
summaries['capex_by_scenario']
```

Out[30]:
```
Scenario
Base        20683800
Upside       3130000
Downside     1590000
Name: Total_Cost_USD, dtype: int64
```

In [31]:
```
summaries['capex_by_scenario'].plot(kind='bar')
plt.ylabel('Total CapEx (USD)')
plt.title('Total CapEx by Scenario')
plt.tight_layout()
plt.show()
```



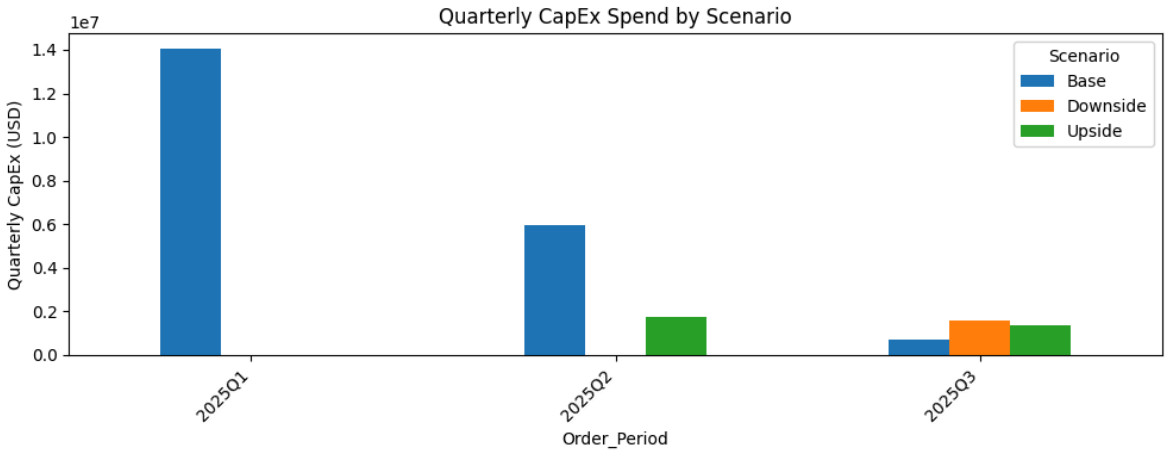
Quarterly CapEx spend curve

In [32]:
```
quarterly_capex = build_quarterly_capex(df)
quarterly_capex.head()
```

Out[32]:

	Scenario	Order_Period	Total_Cost_USD
0	Base	2025Q1	14043800
1	Base	2025Q2	5950000
4	Upside	2025Q2	1770000
2	Base	2025Q3	690000
3	Downside	2025Q3	1590000

In [33]:
```
q_pivot = quarterly_capex.pivot(index='Order_Period', columns='Scenario', values
q_pivot.plot(kind='bar', figsize=(10,4))
plt.ylabel('Quarterly CapEx (USD)')
plt.title('Quarterly CapEx Spend by Scenario')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



Depreciation schedules

In [34]:
```
dep_schedule = build_depreciation_schedule(df)
dep_schedule.head()
```

Out[34]:

	Project_Code	Program_Name	Asset_ID	Year	Annual_Depreciation_USD
0	PRJ-NPI-001	7nm Logic NPI	TOOL-CVD-3000	2025	500000.0
1	PRJ-NPI-001	7nm Logic NPI	TOOL-CVD-3000	2026	500000.0
2	PRJ-NPI-001	7nm Logic NPI	TOOL-CVD-3000	2027	500000.0
3	PRJ-NPI-001	7nm Logic NPI	TOOL-CVD-3000	2028	500000.0
4	PRJ-NPI-001	7nm Logic NPI	TOOL-CVD-3000	2029	500000.0

In [35]:
```
annual_dep_by_program = build_annual_dep_by_program(dep_schedule)
annual_dep_by_program.head()
```

Out[35]:

	Project_Code	Program_Name	Year	Annual_Depreciation_USD
0	PRJ-5S-009	5S & Layout Optimization	2025	4971.428571
1	PRJ-5S-009	5S & Layout Optimization	2026	4971.428571
2	PRJ-5S-009	5S & Layout Optimization	2027	4971.428571
3	PRJ-5S-009	5S & Layout Optimization	2028	4971.428571
4	PRJ-5S-009	5S & Layout Optimization	2029	4971.428571

In [36]:
```
q_dep = build_quarterly_depreciation(df)
q_dep.head()
```

Out[36]:

	Project_Code	Program_Name	Asset_ID	Year	Quarter	Period	Depreciation_USD
0	PRJ-NPI-001	7nm Logic NPI	TOOL-CVD-3000	2025	4	2025Q4	125000.0
1	PRJ-NPI-001	7nm Logic NPI	TOOL-CVD-3000	2026	1	2026Q1	125000.0
2	PRJ-NPI-001	7nm Logic NPI	TOOL-CVD-3000	2026	2	2026Q2	125000.0
3	PRJ-NPI-001	7nm Logic NPI	TOOL-CVD-3000	2026	3	2026Q3	125000.0
4	PRJ-NPI-001	7nm Logic NPI	TOOL-CVD-3000	2026	4	2026Q4	125000.0

Example project cashflows, NPV, IRR

In [37]:
```
example_project = df['Project_Code'].iloc[0]
proj_results = build_project_cashflows(df, example_project)
proj_results
```

Out[37]:
```
{'project_code': 'PRJ-NPI-001',
 'years': [2025, 2026, 2027, 2028, 2029, 2030],
 'cashflows': [np.float64(-3430000.0),
  np.float64(1155600.0),
  np.float64(1155600.0),
  np.float64(1155600.0),
  np.float64(1155600.0),
  np.float64(505600.0)],
 'npv': 547034.330739951,
 'irr': nan,
 'discount_rate': 0.1}
```

In [38]:
```
for year, cf in zip(proj_results['years'], proj_results['cashflows']):
    print(f"Year {year}: {cf:,.0f} USD")
print('\nNPV @ {0:.0%}: {1:,.0f} USD'.format(proj_results['discount_rate'], proj
print('IRR: {0:.2%}'.format(proj_results['irr']) if not np.isnan(proj_results['i
```

```
Year 2025: -3,430,000 USD
Year 2026: 1,155,600 USD
Year 2027: 1,155,600 USD
Year 2028: 1,155,600 USD
Year 2029: 1,155,600 USD
Year 2030: 505,600 USD

NPV @ 10%: 547,034 USD
IRR: N/A
```

Export tables for BI tools (Tableau, Power BI)

In [39]:
```
export_to_csv(quarterly_capex, str(exports_dir / 'quarterly_capex.csv'))
export_to_csv(q_dep, str(exports_dir / 'quarterly_depreciation.csv'))
export_to_csv(annual_dep_by_program, str(exports_dir / 'annual_depreciation_by_p
print('Exports written to', exports_dir)
```

```
Exports written to C:\Users\soura\Documents\CapEx-Forecasting-Engine\exports
```

In [ ]:
```
```