10 K filling report

In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Display all columns
pd.set_option('display.max_columns', None)
```

In [3]:
```
df = pd.read_csv('synthetic_financial_records_15000.csv')
df.head()
```

Out[3]:

	RecordID	Company	Fiscal Year	Total Revenue	Net Income	Total Assets	Li
0	1	Apple	2021	3.720000e+11	9.294797e+10	3.390000e+11	3.060
1	2	Microsoft	2023	2.050000e+11	8.282522e+10	3.700000e+11	2.070
2	3	Tesla	2022	7.571403e+10	1.221922e+10	8.138475e+10	4.179
3	4	Microsoft	2023	2.210000e+11	8.505747e+10	3.730000e+11	2.080
4	5	Microsoft	2022	1.940000e+11	7.407583e+10	3.360000e+11	1.910

In [ ]:
```
# Overview of dataset
df.info()

# Check unique companies and fiscal years
print("Companies:", df['Company'].unique())
print("Years:", df['Fiscal Year'].unique())

#  statistics
df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 10 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   RecordID                              15000 non-null  int64
 1   Company                               15000 non-null  object
 2   Fiscal Year                           15000 non-null  int64
 3   Total Revenue                         15000 non-null  float64
 4   Net Income                            15000 non-null  float64
 5   Total Assets                          15000 non-null  float64
 6   Total Liabilities                     15000 non-null  float64
 7   Cash Flow from Operating Activities   15000 non-null  float64
 8   Source                                15000 non-null  object
 9   Note                                  15000 non-null  object
dtypes: float64(5), int64(2), object(3)
memory usage: 1.1+ MB
Companies: ['Apple' 'Microsoft' 'Tesla']
Years: [2021 2023 2022]
```

Out[ ]:

	RecordID	Fiscal Year	Total Revenue	Net Income	Total Assets	
count	15000.000000	15000.000000	1.500000e+04	1.500000e+04	1.500000e+04	1
mean	7500.500000	2022.002400	2.138809e+11	5.955294e+10	2.527216e+11	1
std	4330.271354	0.816684	1.277114e+11	3.807561e+10	1.274294e+11	1
min	1.000000	2021.000000	4.821528e+10	5.027020e+06	5.522651e+10	2
25%	3750.750000	2021.000000	8.235479e+10	1.467597e+10	8.273345e+10	4
50%	7500.500000	2022.000000	1.970000e+11	7.177137e+10	3.310000e+11	1
75%	11250.250000	2023.000000	3.690000e+11	9.526799e+10	3.520000e+11	2
max	15000.000000	2023.000000	4.410000e+11	1.120000e+11	3.960000e+11	3

In [5]:
```
# Calculate YoY % change for financial metrics by company
df = df.sort_values(by=['Company', 'Fiscal Year'])
df['Revenue Growth (%)'] = df.groupby('Company')['Total Revenue'].pct_change()
df['Net Income Growth (%)'] = df.groupby('Company')['Net Income'].pct_change()
df['Assets Growth (%)'] = df.groupby('Company')['Total Assets'].pct_change() *
df['Liabilities Growth (%)'] = df.groupby('Company')['Total Liabilities'].pct_
df['Cash Flow Growth (%)'] = df.groupby('Company')['Cash Flow from Operating A

# View updated data
df.head(10)
```

Out[5]:

	RecordID	Company	Fiscal Year	Total Revenue	Net Income	Total Assets	I
0	1	Apple	2021	3.720000e+11	9.294797e+10	3.390000e+11	3.06
20	21	Apple	2021	3.610000e+11	9.444124e+10	3.640000e+11	2.75
32	33	Apple	2021	3.730000e+11	9.227637e+10	3.400000e+11	2.91
38	39	Apple	2021	3.670000e+11	9.508108e+10	3.550000e+11	2.71
52	53	Apple	2021	3.560000e+11	9.695245e+10	3.580000e+11	2.68
61	62	Apple	2021	3.680000e+11	9.270741e+10	3.410000e+11	3.00
67	68	Apple	2021	3.640000e+11	9.087138e+10	3.570000e+11	2.96
84	85	Apple	2021	3.440000e+11	9.776107e+10	3.590000e+11	2.85
85	86	Apple	2021	3.720000e+11	9.610104e+10	3.490000e+11	2.85
91	92	Apple	2021	3.690000e+11	1.020000e+11	3.450000e+11	2.85

	RecordID	Company	Fiscal Year	Total Revenue	Net Income	Total Assets	I

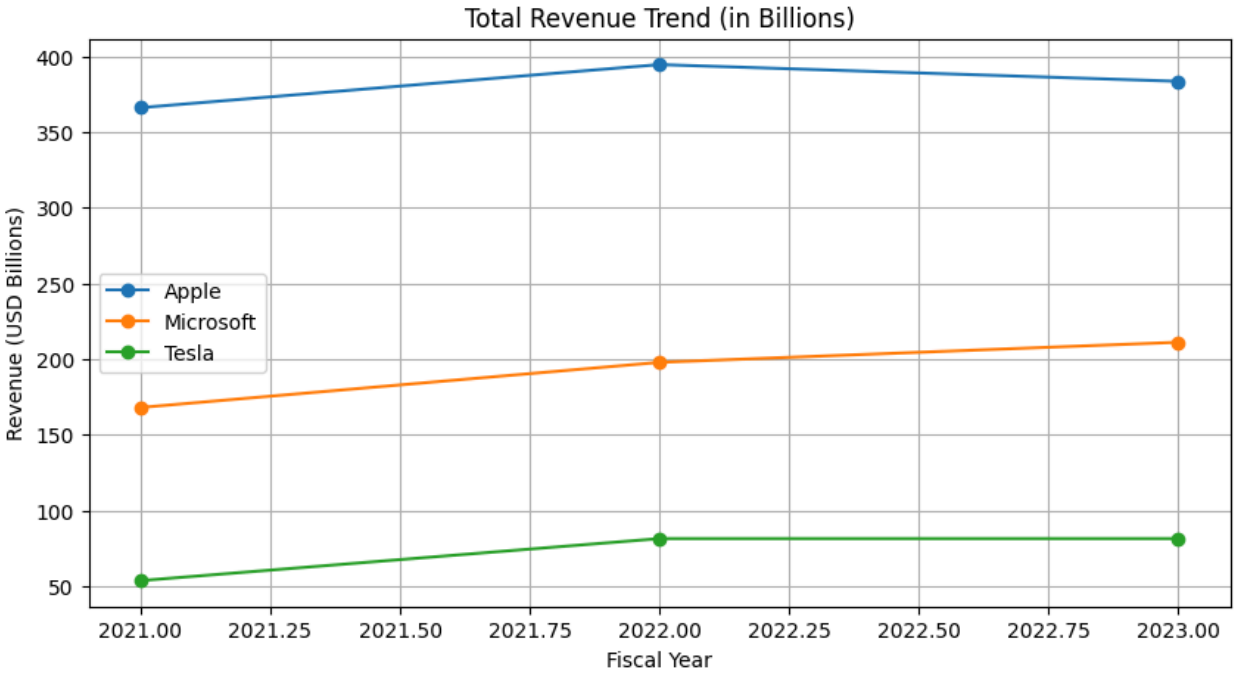
In [12]:
```
# Convert numeric columns to numbers (coerce errors to NaN)
numeric_cols = ['Total Revenue', 'Net Income', 'Total Assets', 'Total Liabilit
df[numeric_cols] = df[numeric_cols].apply(pd.to_numeric, errors='coerce')
```

In [13]:
```
plt.figure(figsize=(10,5))

# Compute the mean only on numeric columns
subset = df.groupby(['Company', 'Fiscal Year'], as_index=False)[['Total Revenu

for company in df['Company'].unique():
    company_data = subset[subset['Company'] == company]
    plt.plot(company_data['Fiscal Year'],
             company_data['Total Revenue'] / 1e9,
             marker='o', label=company)

plt.title('Total Revenue Trend (in Billions)')
plt.xlabel('Fiscal Year')
plt.ylabel('Revenue (USD Billions)')
plt.legend()
plt.grid(True)
plt.show()
```



In [14]:
```
subset = df.groupby(['Company', 'Fiscal Year'], as_index=False).mean(numeric_c
```

Summary of Financial Insights

- **Microsoft** shows consistent revenue growth, with steady increases in net income over the last three years.
- **Tesla** displays high volatility, with rapid expansion from 2021 to 2022, then stable results in 2023.
- **Apple** remains the top performer in total revenue and operating cash flow, though growth slowed slightly in 2023.

These trends suggest:

- Microsoft’s growth is stable and predictable — ideal for long-term investors.
- Tesla remains growth-oriented but volatile.
- Apple leads in cash generation, highlighting strong operational efficiency.

In [ ]:
```
# Save processed DataFrame with growth columns
df.to_csv('financial_analysis_results.csv', index=False)
```

In [22]:
```
%%capture
!jupyter nbconvert --to webpdf --allow-chromium-download "Financial_Analysis.i
```

In [ ]:
```

```