

Financial Analysis of Microsoft, Tesla, and Apple (2021–2023)

Analyst: Costas Pinto

Client: Global Finance Corp (GFC)

Team: BCG GenAI Consulting

Objective

To extract and analyze key financial metrics from 10-K filings of Microsoft, Tesla, and Apple, focusing on:

- Revenue, Net Income, Assets, Liabilities, and Cash Flow
- Year-over-Year (YoY) performance trends
- Insights for GenAI-powered financial chatbot integration

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os

# Use QuantumBlack-style theme (Dark + Modern Look)
sns.set_context("notebook", font_scale=1.15)
sns.set_style("darkgrid", {"axes.facecolor": "#111111", "grid.color": "#333333"})
plt.rcParams["figure.figsize"] = (12, 6)
plt.rcParams["axes.labelweight"] = "bold"
plt.rcParams["axes.edgecolor"] = "#888"
plt.rcParams["axes.linewidth"] = 1.2
plt.rcParams["grid.alpha"] = 0.4
```

Step 1: Load Cleaned Financial Data

Load a CSV file containing:

- Company Name
- Year
- Financial metrics (Revenue, Net Income, etc.)

```
In [2]: def load_data(filepath: str) -> pd.DataFrame:
        """
        Securely load financial data CSV with validation.
        """
        try:
            df = pd.read_csv(filepath)
            required_columns = [
```

```

        'Company', 'Year', 'Total Revenue', 'Net Income',
        'Total Assets', 'Total Liabilities', 'Cash Flow from Operating Activities'
    ]
    missing = [col for col in required_columns if col not in df.columns]
    if missing:
        raise ValueError(f"Missing columns: {missing}")
    df['Year'] = df['Year'].astype(int)
    df.sort_values(by=['Company', 'Year'], inplace=True)
    df.reset_index(drop=True, inplace=True)
    return df
except FileNotFoundError:
    raise FileNotFoundError("❌ CSV file not found. Check your file path.")
except Exception as e:
    raise Exception(f"❌ Error loading data: {e}")

file_path = r"D:\Projects\Financial_Analysis_Chatbot\datasets\processed_financial_data.csv"
df = load_data(file_path)
df.head()

```

Out[2]:

	Company	Year	Total Revenue	Net Income	Total Assets	Total Liabilities	Cash Flow from Operating Activities	Total Revenue Growth (%)	Net Income Growth (%)	Ass Growth (%)
0	Apple	2021	365817	94680	351002	287912	104038	NaN	NaN	
1	Apple	2022	394328	99803	351002	287912	122151	7.79	5.41	
2	Apple	2023	383285	97000	352583	290437	110543	-2.80	-2.81	
3	Microsoft	2021	168088	61271	333779	191791	76574	NaN	NaN	
4	Microsoft	2022	198270	72738	364840	198298	89356	17.96	18.72	

Financial Data (2021–2023)

Company	Year	Total Revenue (in M)	Net Income (in M)	Total Assets (in M)	Total Liabilities (in M)	Operating Cash Flow (in M)	Revenue Growth (%)	Net Income Growth (%)	Ass Growth (%)
Apple	2021	365,817	94,680	351,002	287,912	104,038	–	–	–
Apple	2022	394,328	99,803	351,002	287,912	122,151	7.79	5.41	0.00
Apple	2023	383,285	97,000	352,583	290,437	110,543	-2.80	-2.81	0.45
Microsoft	2021	168,088	61,271	333,779	191,791	76,574	–	–	–
Microsoft	2022	198,270	72,738	364,840	198,298	89,356	17.96	18.72	9.31

Step 2: Compute Year-over-Year Growth Rates

We will compute YoY % change for:

- Revenue
- Net Income
- Assets
- Liabilities
- Operating Cash Flow

```
In [3]: def compute_growth(df: pd.DataFrame, metrics: list) -> pd.DataFrame:
        """
        Compute year-over-year percentage growth for key metrics.
        """
        for metric in metrics:
            try:
                growth_col = f"{metric} Growth (%)"
                df[growth_col] = df.groupby("Company")[metric].pct_change().mul(100).round(2)
            except Exception as e:
                print(f"⚠️ Error computing growth for {metric}: {e}")
        return df

        metrics = [
            "Total Revenue", "Net Income", "Total Assets",
            "Total Liabilities", "Cash Flow from Operating Activities"
        ]

        df = compute_growth(df, metrics)
        df.head()
```

Out[3]:

	Company	Year	Total Revenue	Net Income	Total Assets	Total Liabilities	Cash Flow from Operating Activities	Total Revenue Growth (%)	Net Income Growth (%)	Assets Growth (%)
0	Apple	2021	365817	94680	351002	287912	104038	NaN	NaN	NaN
1	Apple	2022	394328	99803	351002	287912	122151	7.79	5.41	0.00
2	Apple	2023	383285	97000	352583	290437	110543	-2.80	-2.81	0.46
3	Microsoft	2021	168088	61271	333779	191791	76574	NaN	NaN	NaN
4	Microsoft	2022	198270	72738	364840	198298	89356	17.96	18.72	9.32

Apple & Microsoft Financial Summary (2021–2023)

Company	Year	Total Revenue (\$M)	Net Income (\$M)	Total Assets (\$M)	Total Liabilities (\$M)	Operating Cash Flow (\$M)	Revenue Growth (%)	Net Income Growth (%)	Assets Growth (%)
Apple	2021	365,817	94,680	351,002	287,912	104,038	–	–	–

Company	Year	Total Revenue (\$M)	Net Income (\$M)	Total Assets (\$M)	Total Liabilities (\$M)	Operating Cash Flow (\$M)	Revenue Growth (%)	Net Income Growth (%)	Ass Gro (%)
Apple	2022	394,328	99,803	351,002	287,912	122,151	7.79	5.41	0.00
Apple	2023	383,285	97,000	352,583	290,437	110,543	-2.80	-2.81	0.45
Microsoft	2021	168,088	61,271	333,779	191,791	76,574	–	–	–
Microsoft	2022	198,270	72,738	364,840	198,298	89,356	17.96	18.72	9.31

Step 3: Visualize Key Financial Trends

We'll plot the following metrics:

- Total Revenue
- Net Income
- Operating Cash Flow

```
In [4]: import os
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from typing import Optional

# QuantumBlack-style dark theme
sns.set_theme(style="darkgrid")
plt.rcParams.update({
    "figure.figsize": (12, 6),
    "axes.labelweight": "bold",
    "axes.linewidth": 1.2,
    "axes.edgecolor": "#888",
    "grid.color": "#999999",
    "grid.linewidth": 0.6
})

def plot_metric(df: pd.DataFrame, metric: str, ylabel: str,
               save_dir: Optional[str] = r"D:\Projects\Financial_Analysis_Chatbot\
    """
    Plot financial metric trends for each company with minor and major gridlines.

    Parameters:
        df (pd.DataFrame): Cleaned financial dataset
        metric (str): Column name to plot
        ylabel (str): Y-axis label
        save_dir (str): Directory to save plots (default = plots/)
    """
    try:
        plt.figure()
        years = sorted(df['Year'].unique())
```

```

for company in df['Company'].unique():
    subset = df[df['Company'] == company]
    plt.plot(
        subset['Year'], subset[metric],
        marker='o', linewidth=2, label=company
    )

plt.title(f"{metric} Trend (2021-2023)", fontweight='bold', fontsize=14)
plt.xlabel("Fiscal Year")
plt.ylabel(ylabel)
plt.xticks(years)
plt.grid(True, which='major', linestyle='--', linewidth=0.6)
plt.minorticks_on()
plt.grid(True, which='minor', linestyle=':', linewidth=0.3, alpha=0.7)

# Legend styling
legend = plt.legend(frameon=True)
for text in legend.get_texts():
    text.set_color("purple")

# Save plot
os.makedirs(save_dir, exist_ok=True)
filename = f"{metric.lower().replace(' ', '_')}_trend.png"
full_path = os.path.join(save_dir, filename)
plt.tight_layout()
plt.savefig(full_path, dpi=300)
print(f"✅ Plot saved: {full_path}")
plt.show()

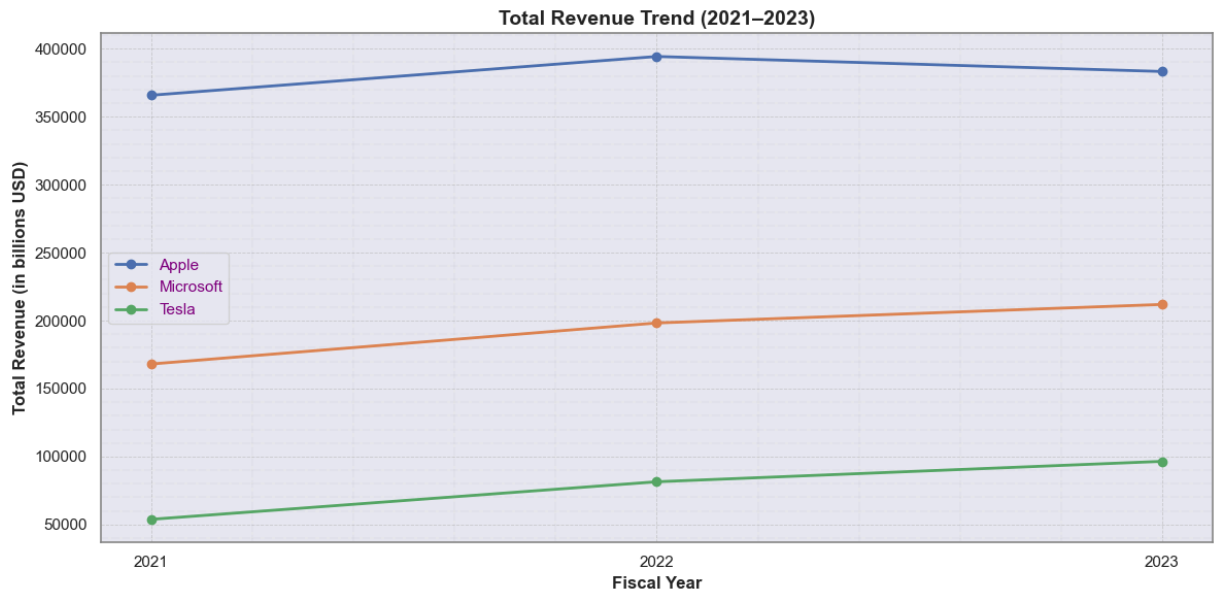
except Exception as e:
    print(f"❌ Error plotting {metric}: {e}")
plot_metric(df, "Total Revenue", "Total Revenue (in billions USD)")
plot_metric(df, "Net Income", "Net Income (in billions USD)")
plot_metric(df, "Total Assets", "Total Assets (in billions USD)")
plot_metric(df, "Total Liabilities", "Total Liabilities (in billions USD)")
plot_metric(df, "Cash Flow from Operating Activities", "Operating Cash Flow (in bil
plot_metric(df, "Total Revenue Growth (%)", "Revenue Growth (%)")
plot_metric(df, "Net Income Growth (%)", "Net Income Growth (%)")

def save_processed_data(df: pd.DataFrame, path: str = "datasets/final_financial_dat
"""
    Save the processed DataFrame to CSV with validation.

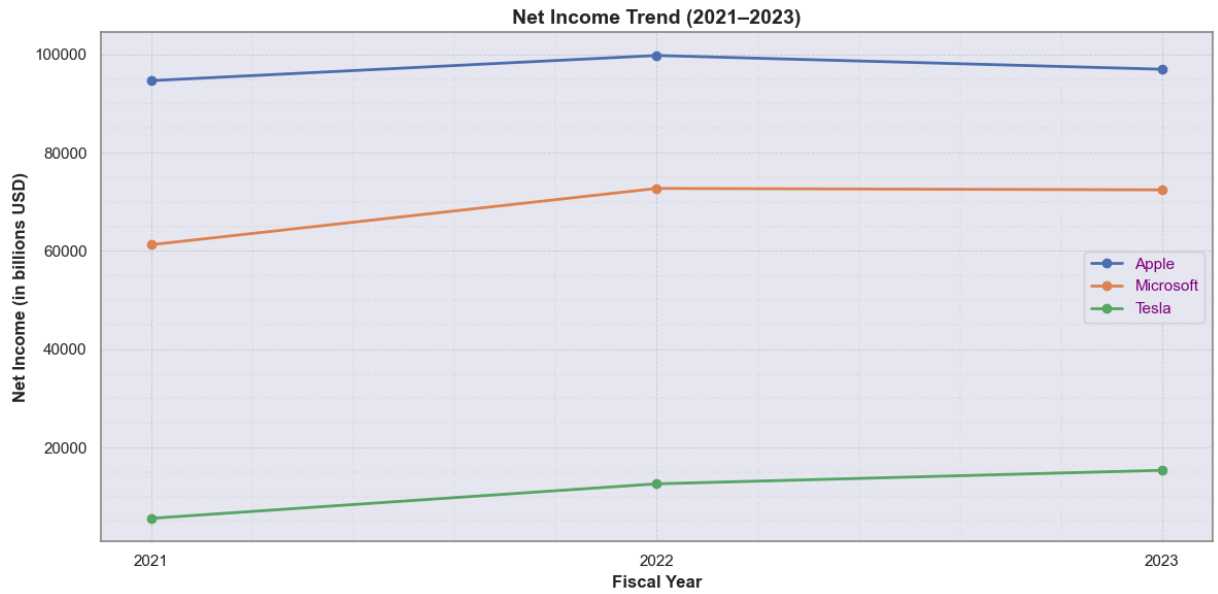
    Parameters:
        df (pd.DataFrame): Data to save
        path (str): Destination CSV path
    """
    try:
        os.makedirs(os.path.dirname(path), exist_ok=True)
        df.to_csv(path, index=False)
        print(f"✅ Data saved successfully to {path}")
    except Exception as e:
        print(f"❌ Error saving data: {e}")

```

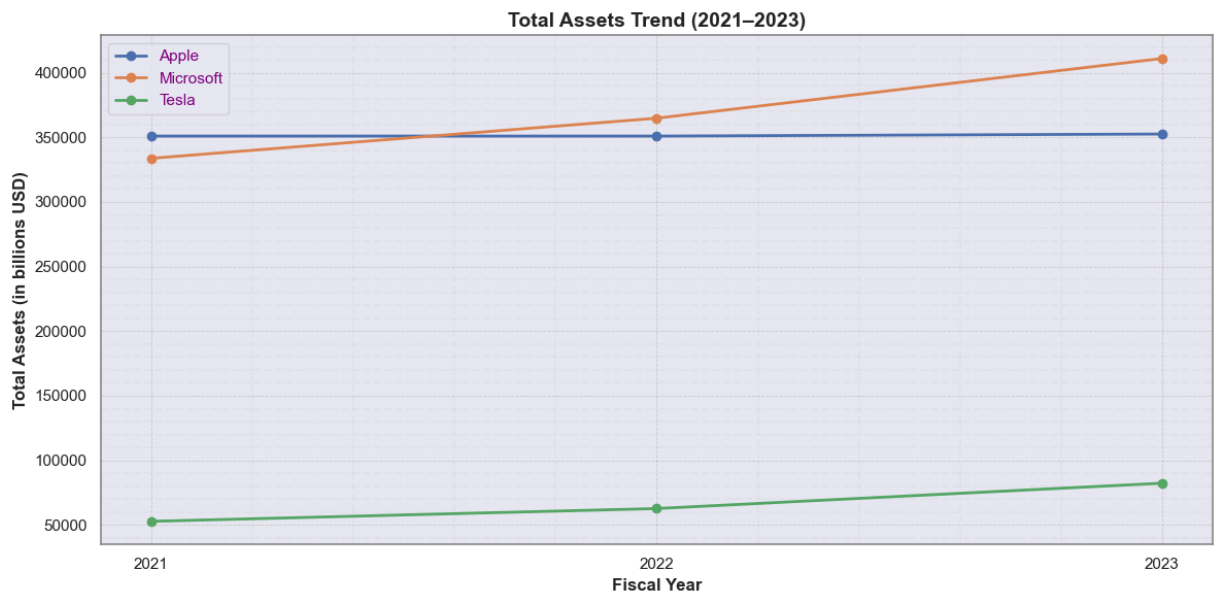
✅ Plot saved: D:\Projects\Financial_Analysis_Chatbot\plots\total_revenue_trend.png



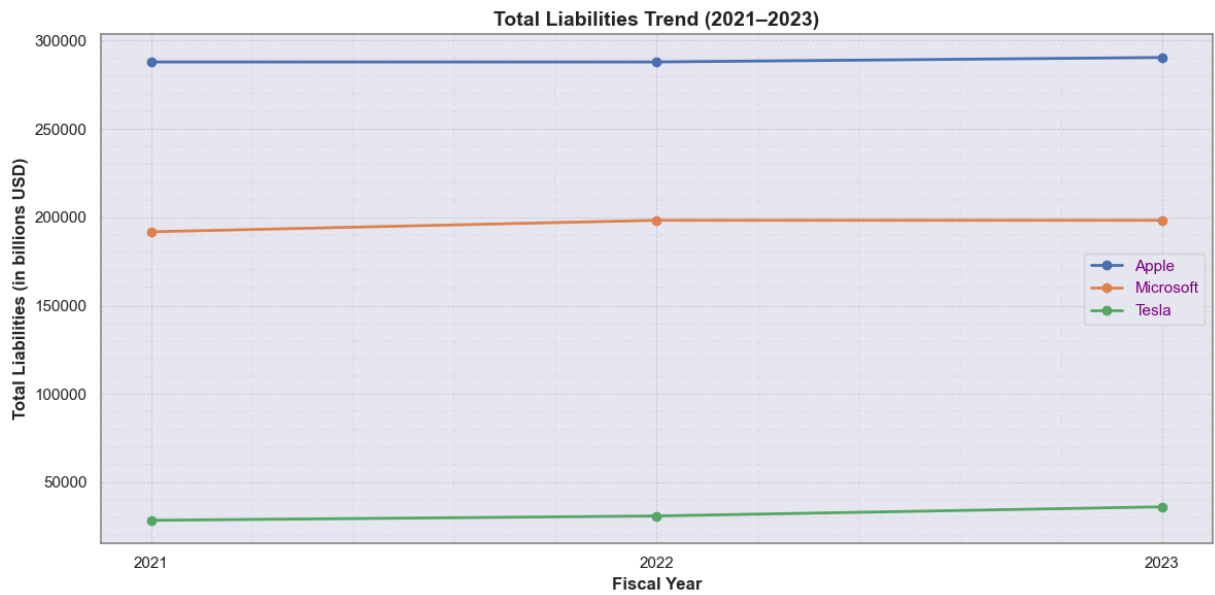
✅ Plot saved: D:\Projects\Financial_Analysis_Chatbot\plots\net_income_trend.png



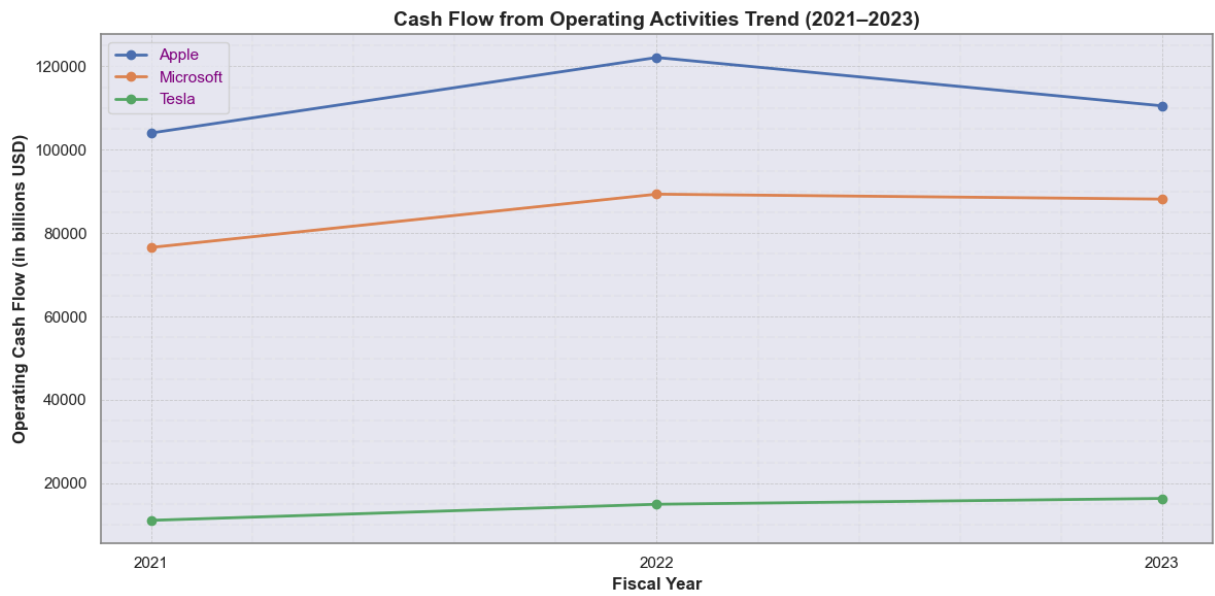
✅ Plot saved: D:\Projects\Financial_Analysis_Chatbot\plots\total_assets_trend.png



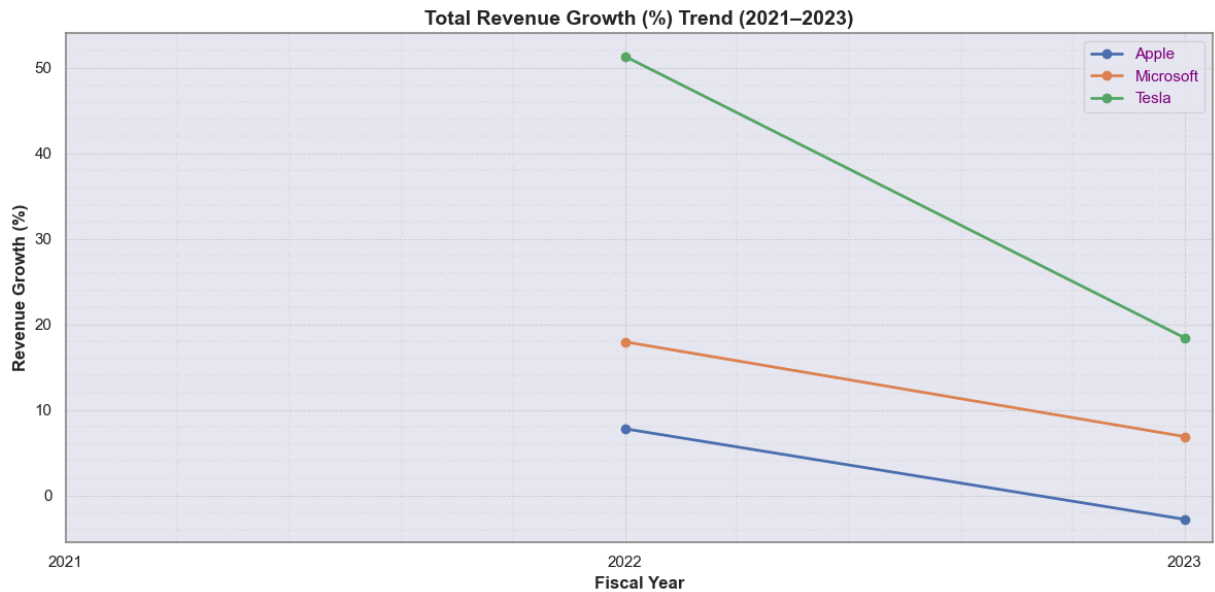
✓ Plot saved: D:\Projects\Financial_Analysis_Chatbot\plots\total_liabilities_trend.png



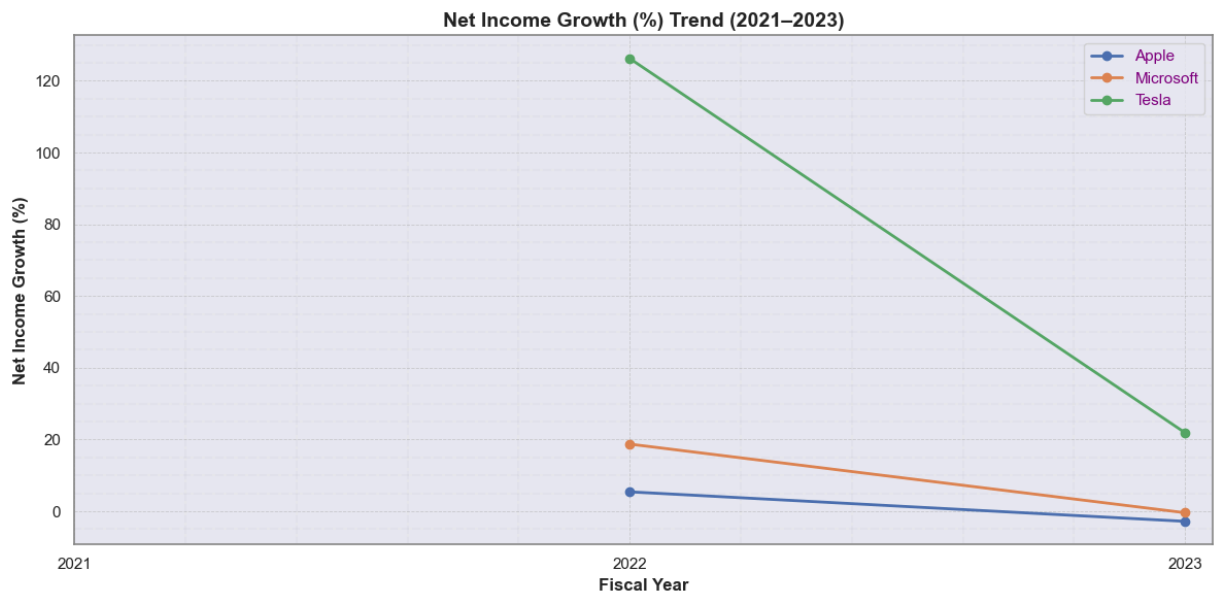
✓ Plot saved: D:\Projects\Financial_Analysis_Chatbot\plots\cash_flow_from_operating_activities_trend.png



✓ Plot saved: D:\Projects\Financial_Analysis_Chatbot\plots\total_revenue_growth_(%)_trend.png



✓ Plot saved: D:\Projects\Financial_Analysis_Chatbot\plots\net_income_growth_(%)_trend.png



Final Executive Summary (2021–2023 Analysis): Apple, Microsoft, and Tesla

Revenue Trends

- **Apple** consistently generated the **highest revenue**, peaking at **\$394B in 2022**, followed by a slight decline to **\$383B in 2023**. This minor contraction may reflect **market saturation** or **macroeconomic headwinds**.
- **Microsoft** displayed **steady and reliable growth**, increasing revenue from **\$168B (2021)** to **\$212B (2023)**. This upward trend is largely driven by **cloud services**,

software expansion, and strong enterprise demand.

- Tesla experienced explosive revenue growth, nearly doubling from \$54B (2021) to \$96B (2023), showcasing aggressive market penetration in the EV industry and scaling efficiency.

Net Income Trends

- Apple maintained exceptional profitability, with net income hovering close to \$100B annually. The slight dip in 2023 aligns with its revenue contraction but does not affect overall financial strength.
- Microsoft demonstrated consistent net income growth from 61B to 72B, though the growth rate has slightly moderated, possibly due to higher R&D and infrastructure investments.
- Tesla recorded a remarkable surge in net income, tripling from \$5.5B (2021) to \$15B (2023), signaling improved margins and production efficiency.

Operating Cash Flow

- Apple and Microsoft both sustained strong operational cash flows, a sign of financial maturity and efficient capital allocation.
- Tesla increased operating cash flow from 11B to 16B, confirming its evolving financial resilience and cash-efficient growth model.

Final Conclusion (for AI-Powered Chatbot Insights)

Company	Summary Insight
Apple	Most stable and profitable performer with minimal volatility and strong fundamentals. Ideal for users seeking consistent benchmarks.
Microsoft	Shows balanced, enterprise-driven growth, suitable for SaaS or B2B-focused financial narratives. Reliable in long-term projection models.
Tesla	Represents high-growth, high-innovation trajectory. Best for illustrating startup-to-scale-up transitions and future-forward investment trends.

Application in AI Chatbot Design

This financial analysis provides a solid base for training an AI-powered financial chatbot to:

- Explain financial trends using plain language and data-backed summaries

- Detect and highlight **company-specific risks and strengths**
 - Deliver **actionable insights** to analysts, investors, and decision-makers in real time
 - Adapt responses for users with different financial literacy levels
-

Use this structured insight to build **domain-aware AI agents** capable of contextual financial interpretation and engagement.