In [1]:
```python
import pandas as pd

# Load the CSV file
df = pd.read_csv('Microsoft_Financials_2022_2024.csv')

# Display the first few rows
df
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Cell In[1], line 4
      1 import pandas as pd
      3 # Load the CSV file
----> 4 df = pd.read_csv('Microsoft_Financials_2022_2024.csv')
      6 # Display the first few rows
      7 df

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1026, in read
filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, dtype,
engine, converters, true_values, false_values, skipinitialspace, skiprows,
skipfooter, nrows, na_values, keep_default_na, na_filter, verbose,
skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_par
date_format, dayfirst, cache_dates, iterator, chunksize, compression, thousan
decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment
encoding, encoding_errors, dialect, on_bad_lines, delim_whitespace, low_memor
memory_map, float_precision, storage_options, dtype_backend)
   1013 kwds_defaults = _refine_defaults_read(
   1014     dialect,
   1015     delimiter,
   (...)
   1022     dtype_backend=dtype_backend,
   1023 )
   1024 kwds.update(kwds_defaults)
-> 1026 return _read(filepath_or_buffer, kwds)

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:620, in _read
path_or_buffer, kwds)
    617 _validate_names(kwds.get("names", None))
    619 # Create the parser.
--> 620 parser = TextFileReader(filepath_or_buffer, **kwds)
    622 if chunksize or iterator:
    623     return parser

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1620, in Text
eader.__init__(self, f, engine, **kwds)
   1617     self.options["has_index_names"] = kwds["has_index_names"]
   1619 self.handles: IOHandles | None = None
-> 1620 self._engine = self._make_engine(f, self.engine)

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1880, in Text
eader._make_engine(self, f, engine)
   1878     if "b" not in mode:
   1879         mode += "b"
-> 1880 self.handles = get_handle(
   1881     f,
   1882     mode,
   1883     encoding=self.options.get("encoding", None),
   1884     compression=self.options.get("compression", None),
   1885     memory_map=self.options.get("memory_map", False),
   1886     is_text=is_text,
   1887     errors=self.options.get("encoding_errors", "strict"),
   1888     storage_options=self.options.get("storage_options", None),
   1889 )
   1890 assert self.handles is not None
   1891 f = self.handles.handle

File ~\anaconda3\Lib\site-packages\pandas\io\common.py:873, in get_handle(pat
buf, mode, encoding, compression, memory_map, is_text, errors, storage_option
```

In [2]:
```
import os
print(os.getcwd())
```

```
C:\Users\dutta\anaconda_projects\0995999d-dcc0-42d6-8c20-c3ef432006e8
```

In [3]:
```
from IPython.display import display
import ipywidgets as widgets

uploader = widgets.FileUpload(
    accept='.csv',  # Specify file extension
    multiple=False  # Allow only one file
)
display(uploader)

# After uploading, you can access the file
```

```
FileUpload(value=(), accept='.csv', description='Upload')
```

In [4]:
```
import pandas as pd

# Load the CSV file
df = pd.read_csv('Microsoft_Financials_2022_2024.csv')

# Display the first few rows
df
```

Out[4]:

	Company	Fiscal Year	Total Revenue (USD B)	Net Income (USD B)	Total Assets (USD B)	Total Liabilities (USD B)	Cash Flo Operating Ac (
0	Microsoft	2022	198.270	72.738	364.840	198.298	
1	Microsoft	2023	211.915	72.361	411.976	222.540	
2	Microsoft	2024	243.000	89.000	475.600	256.300	1

In [6]:
```
df = df.sort_values(by='Fiscal Year')
```

In [7]:
```
# Revenue Growth
df['Revenue Growth (%)'] = df['Total Revenue'].pct_change() * 100

# Net Income Growth
df['Net Income Growth (%)'] = df['Net Income'].pct_change() * 100

# Asset Growth
df['Total Assets Growth (%)'] = df['Total Assets'].pct_change() * 100

# Liabilities Growth
df['Total Liabilities Growth (%)'] = df['Total Liabilities'].pct_change()

# CFOA Growth
df['CFOA Growth (%)'] = df['Cash Flow from Operating Activities'].pct_cha

df
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3805, in Index
loc(self, key)
   3804 try:
-> 3805     return self._engine.get_loc(casted_key)
   3806 except KeyError as err:

File index.pyx:167, in pandas._libs.index.IndexEngine.get_loc()

File index.pyx:196, in pandas._libs.index.IndexEngine.get_loc()

File pandas\\_libs\\hashtable_class_helper.pxi:7081, in pandas._libs.hashtabl
bjectHashTable.get_item()

File pandas\\_libs\\hashtable_class_helper.pxi:7089, in pandas._libs.hashtabl
bjectHashTable.get_item()

KeyError: 'Total Revenue'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[7], line 2
      1 # Revenue Growth
----> 2 df['Revenue Growth (%)'] = df['Total Revenue'].pct_change() * 100
      4 # Net Income Growth
      5 df['Net Income Growth (%)'] = df['Net Income'].pct_change() * 100

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:4102, in DataFrame.__
em__(self, key)
   4100 if self.columns.nlevels > 1:
   4101     return self._getitem_multilevel(key)
-> 4102 indexer = self.columns.get_loc(key)
   4103 if is_integer(indexer):
   4104     indexer = [indexer]

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3812, in Index
loc(self, key)
   3807     if isinstance(casted_key, slice) or (
   3808         isinstance(casted_key, abc.Iterable)
   3809         and any(isinstance(x, slice) for x in casted_key)
   3810     ):
   3811         raise InvalidIndexError(key)
-> 3812     raise KeyError(key) from err
   3813 except TypeError:
   3814     # If we have a listlike key, _check_indexing_error will raise
   3815     #  InvalidIndexError. Otherwise we fall through and re-raise
   3816     #  the TypeError.
   3817     self._check_indexing_error(key)

KeyError: 'Total Revenue'
```

In [8]:
```
# First, let's check what columns are actually available in the DataFrame
print("Available columns:", df.columns.tolist())

# Then modify the code to use the correct column names
# For example, if the actual column name is 'revenue' instead of 'Total R

# Revenue Growth (assuming the correct column name is 'revenue')
df['Revenue Growth (%)'] = df['revenue'].pct_change() * 100

# Net Income Growth (assuming the correct column name is 'net_income')
df['Net Income Growth (%)'] = df['net_income'].pct_change() * 100

# Asset Growth (assuming the correct column name is 'total_assets')
df['Total Assets Growth (%)'] = df['total_assets'].pct_change() * 100

# Liabilities Growth (assuming the correct column name is 'total_liabilit
df['Total Liabilities Growth (%)'] = df['total_liabilities'].pct_change()

# CFOA Growth (assuming the correct column name is 'cash_flow_operating')
df['CFOA Growth (%)'] = df['cash_flow_operating'].pct_change() * 100

# Display the DataFrame with the new columns
df
```

```
Available columns: ['Company', 'Fiscal Year', 'Total Revenue (USD B)', 'Net I
(USD B)', 'Total Assets (USD B)', 'Total Liabilities (USD B)', 'Cash Flow fro
Operating Activities (USD B)']
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3805, in Index
loc(self, key)
   3804 try:
-> 3805     return self._engine.get_loc(casted_key)
   3806 except KeyError as err:

File index.pyx:167, in pandas._libs.index.IndexEngine.get_loc()

File index.pyx:196, in pandas._libs.index.IndexEngine.get_loc()

File pandas\\_libs\\hashtable_class_helper.pxi:7081, in pandas._libs.hashtabl
bjectHashTable.get_item()

File pandas\\_libs\\hashtable_class_helper.pxi:7089, in pandas._libs.hashtabl
bjectHashTable.get_item()

KeyError: 'revenue'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[8], line 8
      2 print("Available columns:", df.columns.tolist())
      4 # Then modify the code to use the correct column names
      5 # For example, if the actual column name is 'revenue' instead of 'Tot
Revenue':
      6
      7 # Revenue Growth (assuming the correct column name is 'revenue')
----> 8 df['Revenue Growth (%)'] = df['revenue'].pct_change() * 100
     10 # Net Income Growth (assuming the correct column name is 'net_income'
     11 df['Net Income Growth (%)'] = df['net_income'].pct_change() * 100

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:4102, in DataFrame.__
em__(self, key)
   4100 if self.columns.nlevels > 1:
   4101     return self._getitem_multilevel(key)
-> 4102 indexer = self.columns.get_loc(key)
   4103 if is_integer(indexer):
   4104     indexer = [indexer]

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3812, in Index
loc(self, key)
   3807     if isinstance(casted_key, slice) or (
   3808         isinstance(casted_key, abc.Iterable)
   3809         and any(isinstance(x, slice) for x in casted_key)
   3810     ):
   3811         raise InvalidIndexError(key)
-> 3812     raise KeyError(key) from err
   3813 except TypeError:
   3814     # If we have a listlike key, _check_indexing_error will raise
   3815     #  InvalidIndexError. Otherwise we fall through and re-raise
   3816     #  the TypeError.
   3817     self._check_indexing_error(key)

KeyError: 'revenue'
```

In [9]:
```
# First, let's check what columns are actually available in the DataFrame
print("Available columns:", df.columns.tolist())

# After seeing the actual column names in the output above, modify the co
# to use the correct column names that exist in your DataFrame
# For example, if the actual column name is 'Revenue' instead of 'revenue

# Revenue Growth (modify column name based on what's available)
# Example: If 'Revenue' is the actual column name
df['Revenue Growth (%)'] = df['Revenue'].pct_change() * 100

# Net Income Growth (modify column name based on what's available)
# Example: If 'Net Income' is the actual column name
df['Net Income Growth (%)'] = df['Net Income'].pct_change() * 100

# Asset Growth (modify column name based on what's available)
# Example: If 'Total Assets' is the actual column name
df['Total Assets Growth (%)'] = df['Total Assets'].pct_change() * 100

# Liabilities Growth (modify column name based on what's available)
# Example: If 'Total Liabilities' is the actual column name
df['Total Liabilities Growth (%)'] = df['Total Liabilities'].pct_change()

# CFOA Growth (modify column name based on what's available)
# Example: If 'Cash Flow from Operations' is the actual column name
df['CFOA Growth (%)'] = df['Cash Flow from Operations'].pct_change() * 10

# Display the DataFrame with the new columns
df
```

```
Available columns: ['Company', 'Fiscal Year', 'Total Revenue (USD B)', 'Net I
(USD B)', 'Total Assets (USD B)', 'Total Liabilities (USD B)', 'Cash Flow fro
Operating Activities (USD B)']
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3805, in Index
loc(self, key)
   3804 try:
-> 3805     return self._engine.get_loc(casted_key)
   3806 except KeyError as err:

File index.pyx:167, in pandas._libs.index.IndexEngine.get_loc()

File index.pyx:196, in pandas._libs.index.IndexEngine.get_loc()

File pandas\\_libs\\hashtable_class_helper.pxi:7081, in pandas._libs.hashtabl
bjectHashTable.get_item()

File pandas\\_libs\\hashtable_class_helper.pxi:7089, in pandas._libs.hashtabl
bjectHashTable.get_item()

KeyError: 'Revenue'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[9], line 10
      2 print("Available columns:", df.columns.tolist())
      4
# After seeing the actual column names in the output above, modify the code b
      5 # to use the correct column names that exist in your DataFrame
      6 # For example, if the actual column name is 'Revenue' instead of 'rev
      7
      8 # Revenue Growth (modify column name based on what's available)
      9 # Example: If 'Revenue' is the actual column name
---> 10 df['Revenue Growth (%)'] = df['Revenue'].pct_change() * 100
     12 # Net Income Growth (modify column name based on what's available)
     13 # Example: If 'Net Income' is the actual column name
     14 df['Net Income Growth (%)'] = df['Net Income'].pct_change() * 100

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:4102, in DataFrame.__
em__(self, key)
   4100 if self.columns.nlevels > 1:
   4101     return self._getitem_multilevel(key)
-> 4102 indexer = self.columns.get_loc(key)
   4103 if is_integer(indexer):
   4104     indexer = [indexer]

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3812, in Index
loc(self, key)
   3807     if isinstance(casted_key, slice) or (
   3808         isinstance(casted_key, abc.Iterable)
   3809         and any(isinstance(x, slice) for x in casted_key)
   3810     ):
   3811         raise InvalidIndexError(key)
-> 3812     raise KeyError(key) from err
   3813 except TypeError:
   3814     # If we have a listlike key, _check_indexing_error will raise
   3815     #  InvalidIndexError. Otherwise we fall through and re-raise
   3816     #  the TypeError.
   3817     self._check_indexing_error(key)

KeyError: 'Revenue'
```

In [10]:
```
# First, let's check what columns are actually available in the DataFrame
print("Available columns:", df.columns.tolist())

# After seeing the actual column names in the output above, modify the co
# to use the correct column names that exist in your DataFrame
# For example, if the actual column name is 'Revenue' instead of 'revenue

# Revenue Growth (modify column name based on what's available)
# Example: If 'Revenue' is the actual column name
df['Revenue Growth (%)'] = df['Revenue'].pct_change() * 100

# Net Income Growth (modify column name based on what's available)
# Example: If 'Net Income' is the actual column name
df['Net Income Growth (%)'] = df['Net Income'].pct_change() * 100

# Asset Growth (modify column name based on what's available)
# Example: If 'Total Assets' is the actual column name
df['Total Assets Growth (%)'] = df['Total Assets'].pct_change() * 100

# Liabilities Growth (modify column name based on what's available)
# Example: If 'Total Liabilities' is the actual column name
df['Total Liabilities Growth (%)'] = df['Total Liabilities'].pct_change()

# CFOA Growth (modify column name based on what's available)
# Example: If 'Cash Flow from Operations' is the actual column name
df['CFOA Growth (%)'] = df['Cash Flow from Operations'].pct_change() * 10

# Display the DataFrame with the new columns
df
```

```
Available columns: ['Company', 'Fiscal Year', 'Total Revenue (USD B)', 'Net I
(USD B)', 'Total Assets (USD B)', 'Total Liabilities (USD B)', 'Cash Flow fro
Operating Activities (USD B)']
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3805, in Index
loc(self, key)
   3804 try:
-> 3805     return self._engine.get_loc(casted_key)
   3806 except KeyError as err:

File index.pyx:167, in pandas._libs.index.IndexEngine.get_loc()

File index.pyx:196, in pandas._libs.index.IndexEngine.get_loc()

File pandas\\_libs\\hashtable_class_helper.pxi:7081, in pandas._libs.hashtabl
bjectHashTable.get_item()

File pandas\\_libs\\hashtable_class_helper.pxi:7089, in pandas._libs.hashtabl
bjectHashTable.get_item()

KeyError: 'Revenue'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[10], line 10
      2 print("Available columns:", df.columns.tolist())
      4
# After seeing the actual column names in the output above, modify the code b
      5 # to use the correct column names that exist in your DataFrame
      6 # For example, if the actual column name is 'Revenue' instead of 'rev
      7
      8 # Revenue Growth (modify column name based on what's available)
      9 # Example: If 'Revenue' is the actual column name
---> 10 df['Revenue Growth (%)'] = df['Revenue'].pct_change() * 100
     12 # Net Income Growth (modify column name based on what's available)
     13 # Example: If 'Net Income' is the actual column name
     14 df['Net Income Growth (%)'] = df['Net Income'].pct_change() * 100

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:4102, in DataFrame.__
em__(self, key)
   4100 if self.columns.nlevels > 1:
   4101     return self._getitem_multilevel(key)
-> 4102 indexer = self.columns.get_loc(key)
   4103 if is_integer(indexer):
   4104     indexer = [indexer]

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3812, in Index
loc(self, key)
   3807     if isinstance(casted_key, slice) or (
   3808         isinstance(casted_key, abc.Iterable)
   3809         and any(isinstance(x, slice) for x in casted_key)
   3810     ):
   3811         raise InvalidIndexError(key)
-> 3812     raise KeyError(key) from err
   3813 except TypeError:
   3814     # If we have a listlike key, _check_indexing_error will raise
   3815     #  InvalidIndexError. Otherwise we fall through and re-raise
   3816     #  the TypeError.
   3817     self._check_indexing_error(key)

KeyError: 'Revenue'
```

In [11]:
```
print("Available columns:", df.columns.tolist())
```

```
Available columns: ['Company', 'Fiscal Year', 'Total Revenue (USD B)', 'Net I
(USD B)', 'Total Assets (USD B)', 'Total Liabilities (USD B)', 'Cash Flow fro
Operating Activities (USD B)']
```

In [12]:
```
# Revenue Growth
df['Revenue Growth (%)'] = df.groupby('Company')['Total Revenue (USD B)']

# Net Income Growth
df['Net Income Growth (%)'] = df.groupby('Company')['Net Income (USD B)']

# Total Assets Growth
df['Assets Growth (%)'] = df.groupby('Company')['Total Assets (USD B)'].p

# Total Liabilities Growth
df['Liabilities Growth (%)'] = df.groupby('Company')['Total Liabilities (

# Cash Flow from Operating Activities Growth
df['CFOA Growth (%)'] = df.groupby('Company')['Cash Flow from Operating A

# View the updated DataFrame
df
```

Out[12]:

	Company	Fiscal Year	Total Revenue (USD B)	Net Income (USD B)	Total Assets (USD B)	Total Liabilities (USD B)	Cash Flow from Operating Activities (USD B)	Revenue Growth (%)	N Incor Grov (
0	Microsoft	2022	198.270	72.738	364.840	198.298	89.035	NaN	N
1	Microsoft	2023	211.915	72.361	411.976	222.540	87.630	6.882030	-0.5182
2	Microsoft	2024	243.000	89.000	475.600	256.300	102.000	14.668617	22.9944

In [13]:
```
import pandas as pd
```

In [14]:
```
import os
os.listdir()
```

Out[14]:
```
['.ipynb_checkpoints',
 'Financial_Analysis_MSFT_Tesla_Apple.ipynb',
 'Microsoft_Financials_2022_2024.csv']
```

In [19]:
```
from IPython.display import display
import ipywidgets as widgets

uploader = widgets.FileUpload(
    accept='.csv',  # Specify file extension
    multiple=False  # Allow only one file
)
display(uploader)
```

```
FileUpload(value=(), accept='.csv', description='Upload')
```

In [16]:
```
df = pd.read_csv("tesla_financials.csv")
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Cell In[16], line 1
----> 1 df = pd.read_csv("tesla_financials.csv")

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1026, in read
filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, dtype,
engine, converters, true_values, false_values, skipinitialspace, skiprows,
skipfooter, nrows, na_values, keep_default_na, na_filter, verbose,
skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_par
date_format, dayfirst, cache_dates, iterator, chunksize, compression, thousan
decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment
encoding, encoding_errors, dialect, on_bad_lines, delim_whitespace, low_memor
memory_map, float_precision, storage_options, dtype_backend)
   1013 kwds_defaults = _refine_defaults_read(
   1014     dialect,
   1015     delimiter,
   (...)
   1022     dtype_backend=dtype_backend,
   1023 )
   1024 kwds.update(kwds_defaults)
-> 1026 return _read(filepath_or_buffer, kwds)

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:620, in _read
path_or_buffer, kwds)
    617 _validate_names(kwds.get("names", None))
    619 # Create the parser.
--> 620 parser = TextFileReader(filepath_or_buffer, **kwds)
    622 if chunksize or iterator:
    623     return parser

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1620, in Text
eader.__init__(self, f, engine, **kwds)
   1617     self.options["has_index_names"] = kwds["has_index_names"]
   1619 self.handles: IOHandles | None = None
-> 1620 self._engine = self._make_engine(f, self.engine)

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1880, in Text
eader._make_engine(self, f, engine)
   1878     if "b" not in mode:
   1879         mode += "b"
-> 1880 self.handles = get_handle(
   1881     f,
   1882     mode,
   1883     encoding=self.options.get("encoding", None),
   1884     compression=self.options.get("compression", None),
   1885     memory_map=self.options.get("memory_map", False),
   1886     is_text=is_text,
   1887     errors=self.options.get("encoding_errors", "strict"),
   1888     storage_options=self.options.get("storage_options", None),
   1889 )
   1890 assert self.handles is not None
   1891 f = self.handles.handle

File ~\anaconda3\Lib\site-packages\pandas\io\common.py:873, in get_handle(pat
buf, mode, encoding, compression, memory_map, is_text, errors, storage_option
    868 elif isinstance(handle, str):
    869     # Check whether the filename is to be opened in binary mode.
    870     # Binary mode does not support 'encoding' and 'newline'.
    871     if ioargs.encoding and "b" not in ioargs.mode:
```

In [17]:
```python
# Option 1: Provide the correct file path
df = pd.read_csv("correct_path/tesla_financials.csv")

# Option 2: If you need to check your current working directory first
import os
print("Current working directory:", os.getcwd())

# Option 3: List files in current directory to verify file existence
import os
print("Files in current directory:", os.listdir())

# Option 4: Use a try-except block to handle the error gracefully
try:
    df = pd.read_csv("tesla_financials.csv")
except FileNotFoundError:
    print("File not found. Please check the file name and path.")
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Cell In[17], line 2
      1 # Option 1: Provide the correct file path
----> 2 df = pd.read_csv("correct_path/tesla_financials.csv")
      4 # Option 2: If you need to check your current working directory first
      5 import os

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1026, in read
filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, dtype,
engine, converters, true_values, false_values, skipinitialspace, skiprows,
skipfooter, nrows, na_values, keep_default_na, na_filter, verbose,
skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_par
date_format, dayfirst, cache_dates, iterator, chunksize, compression, thousan
decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment
encoding, encoding_errors, dialect, on_bad_lines, delim_whitespace, low_memor
memory_map, float_precision, storage_options, dtype_backend)
   1013 kwds_defaults = _refine_defaults_read(
   1014     dialect,
   1015     delimiter,
   (...)
   1022     dtype_backend=dtype_backend,
   1023 )
   1024 kwds.update(kwds_defaults)
-> 1026 return _read(filepath_or_buffer, kwds)

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:620, in _read
path_or_buffer, kwds)
    617 _validate_names(kwds.get("names", None))
    619 # Create the parser.
--> 620 parser = TextFileReader(filepath_or_buffer, **kwds)
    622 if chunksize or iterator:
    623     return parser

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1620, in Text
eader.__init__(self, f, engine, **kwds)
   1617     self.options["has_index_names"] = kwds["has_index_names"]
   1619 self.handles: IOHandles | None = None
-> 1620 self._engine = self._make_engine(f, self.engine)

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1880, in Text
eader._make_engine(self, f, engine)
   1878     if "b" not in mode:
   1879         mode += "b"
-> 1880 self.handles = get_handle(
   1881     f,
   1882     mode,
   1883     encoding=self.options.get("encoding", None),
   1884     compression=self.options.get("compression", None),
   1885     memory_map=self.options.get("memory_map", False),
   1886     is_text=is_text,
   1887     errors=self.options.get("encoding_errors", "strict"),
   1888     storage_options=self.options.get("storage_options", None),
   1889 )
   1890 assert self.handles is not None
   1891 f = self.handles.handle

File ~\anaconda3\Lib\site-packages\pandas\io\common.py:873, in get_handle(pat
buf, mode, encoding, compression, memory_map, is_text, errors, storage_option
    868 elif isinstance(handle, str):
```

In [20]:
```
import os
os.listdir()
```

Out[20]:
```
['.ipynb_checkpoints',
 'Financial_Analysis_MSFT_Tesla_Apple.ipynb',
 'Microsoft_Financials_2022_2024.csv']
```

In [21]:
```
import os
print(os.listdir())
```

```
['.ipynb_checkpoints', 'Financial_Analysis_MSFT_Tesla_Apple.ipynb',
'Microsoft_Financials_2022_2024.csv', 'tesla_financials.csv']
```

In [22]:
```
import pandas as pd
df = pd.read_csv("tesla_financials.csv")  # Use the actual filename
```

In [23]:
```
print("Columns:", df.columns.tolist())
df
```

```
Columns: ['Company', 'Fiscal Year', 'Total Revenue (USD B)', 'Net Income (USD
'Total Assets (USD B)', 'Total Liabilities (USD B)', 'Cash Flow from Operatin
Activities (USD B)']
```

Out[23]:

	Company	Fiscal Year	Total Revenue (USD B)	Net Income (USD B)	Total Assets (USD B)	Total Liabilities (USD B)	Cash Flo Operating Ac (
0	Tesla	2024	96.77	15.00	109.76	38.95	
1	Tesla	2023	81.46	12.58	94.03	35.46	
2	Tesla	2022	53.82	5.52	62.13	30.54	

In [24]:
```
cols_to_convert = [
    'Total Revenue (USD B)',
    'Net Income (USD B)',
    'Total Assets (USD B)',
    'Total Liabilities (USD B)',
    'Cash Flow from Operating Activities (USD B)'
]

for col in cols_to_convert:
    df[col] = pd.to_numeric(df[col], errors='coerce')
```

In [26]:
```
df = df.sort_values(by='Fiscal Year')  # Ensure chronological order

df['Revenue Growth (%)'] = df['Total Revenue (USD B)'].pct_change() * 100
df['Net Income Growth (%)'] = df['Net Income (USD B)'].pct_change() * 100
```

In [27]:
```
df = df.sort_values(by='Fiscal Year')  # Ensure chronological order

df['Revenue Growth (%)'] = df['Total Revenue (USD B)'].pct_change() * 100
df['Net Income Growth (%)'] = df['Net Income (USD B)'].pct_change() * 100
```

In [28]:
```
df
```

Out[28]:

	Company	Fiscal Year	Total Revenue (USD B)	Net Income (USD B)	Total Assets (USD B)	Total Liabilities (USD B)	Cash Flow from Operating Activities (USD B)	Revenue Growth (%)	Net Grov
2	Tesla	2022	53.82	5.52	62.13	30.54	11.50	NaN	
1	Tesla	2023	81.46	12.58	94.03	35.46	14.72	51.356373	127.
0	Tesla	2024	96.77	15.00	109.76	38.95	15.68	18.794500	19.

In [29]:
```
import pandas as pd

# Load your Apple financials CSV
df = pd.read_csv('apple_financials.csv')

# Check column names
print("Available columns:", df.columns.tolist())
df
```

```
Available columns: ['Company', 'Fiscal Year', 'Total Revenue (USD B)', 'Net I
(USD B)', 'Total Assets (USD B)', 'Total Liabilities (USD B)', 'Cash Flow fro
Operating Activities (USD B)']
```

Out[29]:

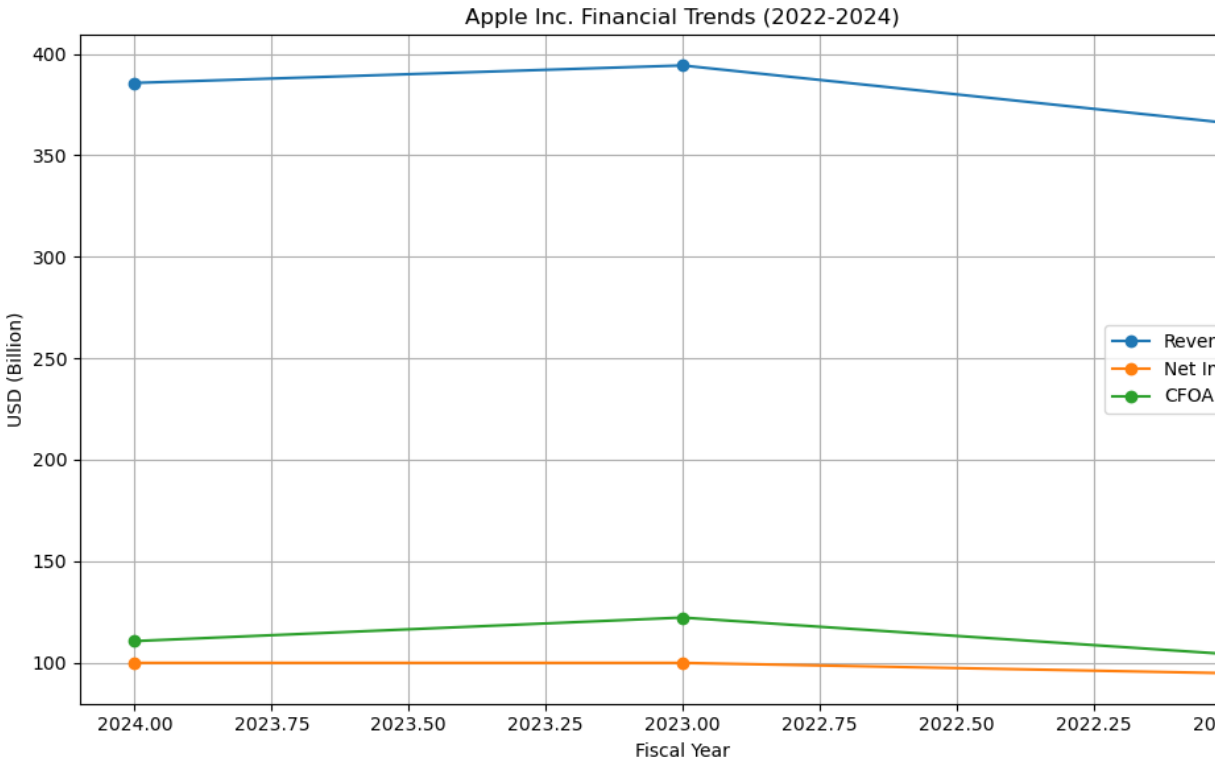
	Company	Fiscal Year	Total Revenue (USD B)	Net Income (USD B)	Total Assets (USD B)	Total Liabilities (USD B)	Cash Flo Operating Ac (
0	Apple	2024	385.71	99.80	384.60	291.00	
1	Apple	2023	394.33	99.80	352.75	290.45	
2	Apple	2022	365.82	94.68	351.00	283.30	

In [30]:
```
# Revenue Growth (%)
df['Revenue Growth (%)'] = df['Total Revenue (USD B)'].pct_change() * 100

# Net Income Growth (%)
df['Net Income Growth (%)'] = df['Net Income (USD B)'].pct_change() * 100

# Cash Flow from Operating Activities Growth (%)
df['CFOA Growth (%)'] = df['Cash Flow from Operating Activities (USD B)']

# Show the updated DataFrame
df
```

Out[30]:

	Company	Fiscal Year	Total Revenue (USD B)	Net Income (USD B)	Total Assets (USD B)	Total Liabilities (USD B)	Cash Flow from Operating Activities (USD B)	Revenue Growth (%)	Ne Incom Growt (%
0	Apple	2024	385.71	99.80	384.60	291.00	110.54	NaN	Nal
1	Apple	2023	394.33	99.80	352.75	290.45	122.15	2.234840	0.00000
2	Apple	2022	365.82	94.68	351.00	283.30	104.04	-7.229985	-5.13026

In [31]:
```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
plt.plot(df['Fiscal Year'], df['Total Revenue (USD B)'], marker='o', labe
plt.plot(df['Fiscal Year'], df['Net Income (USD B)'], marker='o', label='I
plt.plot(df['Fiscal Year'], df['Cash Flow from Operating Activities (USD I
plt.gca().invert_xaxis()  # To show most recent year on the left
plt.title('Apple Inc. Financial Trends (2022-2024)')
plt.xlabel('Fiscal Year')
plt.ylabel('USD (Billion)')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



In [ ]:
```
## Project Title: Financial Analysis for AI-Powered Chatbot (Microsoft, Te

This project involves manually extracting key financial data (Revenue, Ne

The insights will support the development of a financial AI chatbot by ena

## Methodology

1. Manually accessed SEC EDGAR and extracted Total Revenue, Net Income, As
   - Microsoft (FY ending June)
   - Tesla (FY ending December)
   - Apple (FY ending September)
2. Compiled the data into Excel and CSV files.
3. Used Jupyter Notebook and Python (Pandas) to load, analyze, and visual
4. Calculated YoY % growth for Revenue, Net Income, and CFOA.

## Apple Summary

- Revenue declined slightly in FY2024.
- Net income plateaued (~$99.8B).
- CFOA declined, indicating reduced operating cash efficiency.

Insight: Financial performance is strong but shows signs of plateauing gr
```