In [43]:

```
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [44]:

```
indices = ['^GSPC', '^DJI', '^IXIC', '^FTSE', '^GDAXI']
equities = ['AAPL', 'MSFT', 'AMZN', 'GOOGL', 'JNJ']
```

In [45]:

```
start_date = '2010-01-01'
end_date = '2023-05-01'

index_data = yf.download(indices, start=start_date, end=end_date)['Adj Close']
equity_data = yf.download(equities, start=start_date, end=end_date)['Adj Close']
```

```
[**********100%**********]  5 of 5 completed
[**********100%**********]  5 of 5 completed
```

In [46]:

```
index_data
```

Out[46]:

Date	^DJI	^FTSE	^GDAXI	^GSPC	^IXIC
2010-01-04	10583.959961	5500.299805	6048.299805	1132.989990	2308.419922
2010-01-05	10572.019531	5522.500000	6031.859863	1136.520020	2308.709961
2010-01-06	10573.679688	5530.000000	6034.330078	1137.140015	2301.090088
2010-01-07	10606.860352	5526.700195	6019.359863	1141.689941	2300.050049
2010-01-08	10618.190430	5534.200195	6037.609863	1144.979980	2317.169922
...	...	...	...	...	...
2023-04-24	33875.398438	7912.200195	15863.950195	4137.040039	12037.200195
2023-04-25	33530.828125	7891.100098	15872.129883	4071.629883	11799.160156
2023-04-26	33301.871094	7852.600098	15795.730469	4055.989990	11854.349609
2023-04-27	33826.160156	7831.600098	15800.450195	4135.350098	12142.240234
2023-04-28	34098.160156	7870.600098	15922.379883	4169.479980	12226.580078

3441 rows × 5 columns

In [47]:

```
equity_data
```

Out[47]:

Date	AAPL	AMZN	GOOGL	JNJ	MSFT
2010-01-04	6.496294	6.695000	15.684434	43.962326	23.572369
2010-01-05	6.507526	6.734500	15.615365	43.452568	23.579988
2010-01-06	6.404015	6.612500	15.221722	43.806000	23.435282
2010-01-07	6.392175	6.500000	14.867367	43.493340	23.191553
2010-01-08	6.434673	6.676000	15.065566	43.642879	23.351496
...	...	...	...	...	...
2023-04-24	165.101639	106.209999	105.970001	163.679993	281.155365
2023-04-25	163.543793	102.570000	103.849998	165.179993	274.819244
2023-04-26	163.533798	104.980003	103.709999	162.619995	294.725708
2023-04-27	168.177383	109.820000	107.589996	163.000000	304.165070
2023-04-28	169.445618	105.449997	107.339996	163.699997	306.589783

3353 rows × 5 columns

In [48]:

```
index_returns=index_data.pct_change().dropna()
index_returns
```

Out[48]:

Date	^DJI	^FTSE	^GDAXI	^GSPC	^IXIC
2010-01-05	-0.001128	0.004036	-0.002718	0.003116	0.000126
2010-01-06	0.000157	0.001358	0.000410	0.000546	-0.003300
2010-01-07	0.003138	-0.000597	-0.002481	0.004001	-0.000452
2010-01-08	0.001068	0.001357	0.003032	0.002882	0.007443
2010-01-11	0.004313	0.000705	0.000479	0.001747	-0.002054
...	...	...	...	...	...
2023-04-24	0.001965	-0.000240	-0.001115	0.000852	-0.002921
2023-04-25	-0.010172	-0.002667	0.000516	-0.015811	-0.019775
2023-04-26	-0.006828	-0.004879	-0.004813	-0.003841	0.004677
2023-04-27	0.015744	-0.002674	0.000299	0.019566	0.024286
2023-04-28	0.008041	0.004980	0.007717	0.008253	0.006946

3440 rows × 5 columns

In [49]:

```
equity_returns=equity_data.pct_change().dropna()
equity_returns
```

Out[49]:

Date	AAPL	AMZN	GOOGL	JNJ	MSFT
2010-01-05	0.001729	0.005900	-0.004404	-0.011595	0.000323
2010-01-06	-0.015906	-0.018116	-0.025209	0.008134	-0.006137
2010-01-07	-0.001849	-0.017013	-0.023280	-0.007137	-0.010400
2010-01-08	0.006648	0.027077	0.013331	0.003438	0.006897
2010-01-11	-0.008822	-0.024041	-0.001512	0.000156	-0.012720
...	...	...	...	...	...
2023-04-24	0.001879	-0.007012	0.005313	0.006085	-0.013963
2023-04-25	-0.009436	-0.034272	-0.020006	0.009164	-0.022536
2023-04-26	-0.000061	0.023496	-0.001348	-0.015498	0.072435
2023-04-27	0.028395	0.046104	0.037412	0.002337	0.032028
2023-04-28	0.007541	-0.039792	-0.002324	0.004294	0.007972

3352 rows × 5 columns

In [50]:

```
index_cumulative_returns=(1+index_returns).cumprod()-1
equity_cumulative_returns=(1+equity_returns).cumprod()-1
index_cumulative_returns
```

Out[50]:

Date	^DJI	^FTSE	^GDAXI	^GSPC	^IXIC
2010-01-05	-0.001128	0.004036	-0.002718	0.003116	0.000126
2010-01-06	-0.000971	0.005400	-0.002310	0.003663	-0.003175
2010-01-07	0.002164	0.004800	-0.004785	0.007679	-0.003626
2010-01-08	0.003234	0.006163	-0.001767	0.010583	0.003790
2010-01-11	0.007561	0.006872	-0.001290	0.012348	0.001728
...	...	...	...	...	...
2023-04-24	2.200636	0.438503	1.622878	2.651436	4.214476
2023-04-25	2.168080	0.434667	1.624230	2.593703	4.111358
2023-04-26	2.146447	0.427668	1.611598	2.579899	4.135266
2023-04-27	2.195983	0.423850	1.612379	2.649944	4.259979
2023-04-28	2.221683	0.430940	1.632538	2.680068	4.296515

3440 rows × 5 columns

In [51]:

```
equity_cumulative_returns
```

Out[51]:

Date	AAPL	AMZN	GOOGL	JNJ	MSFT
2010-01-05	0.001729	0.005900	-0.004404	-0.011595	0.000323
2010-01-06	-0.014205	-0.012323	-0.029501	-0.003556	-0.005816
2010-01-07	-0.016027	-0.029126	-0.052094	-0.010668	-0.016155
2010-01-08	-0.009486	-0.002838	-0.039457	-0.007266	-0.009370
2010-01-11	-0.018223	-0.026811	-0.040909	-0.007112	-0.021971
...	...	...	...	...	...
2023-04-24	24.414742	14.864077	5.756380	2.723188	10.927328
2023-04-25	24.174937	14.320388	5.621214	2.757308	10.658533
2023-04-26	24.173399	14.680359	5.612288	2.699076	11.503016
2023-04-27	24.888204	15.403286	5.859667	2.707720	11.903458
2023-04-28	25.083428	14.750559	5.843728	2.723643	12.006321

3352 rows × 5 columns

In [52]:

```
index_rolling_max = index_data.rolling(window=len(index_data), min_periods=1).max()
index_drawdown = (index_data - index_rolling_max) / (index_rolling_max)
index_max_drawdown = index_drawdown.min()
index_max_drawdown
```

Out[52]:

```
^DJI     -0.370862
^FTSE    -0.366055
^GDAXI   -0.387794
^GSPC    -0.339250
^IXIC    -0.363953
dtype: float64
```

In [53]:

```
equity_rolling_max = equity_data.rolling(window=len(equity_data), min_periods=1).max()
equity_drawdown = (equity_data - equity_rolling_max) / (equity_rolling_max)
equity_max_drawdown = equity_drawdown.min()
equity_max_drawdown
```

Out[53]:

```
AAPL    -0.437972
AMZN    -0.561453
GOOGL   -0.443201
JNJ     -0.273663
MSFT    -0.371485
dtype: float64
```

In [54]:

```
index_volatility = index_returns.std()
equity_volatility = equity_returns.std()
index_volatility
```

Out[54]:

```
^DJI      0.010684
^FTSE     0.010121
^GDAXI    0.012649
^GSPC     0.011044
^IXIC     0.012813
dtype: float64
```

In [55]:

```
equity_volatility
```

Out[55]:

```
AAPL     0.018016
AMZN     0.020901
GOOGL    0.017196
JNJ      0.010618
MSFT     0.016466
dtype: float64
```

In [56]:

```
risk_free_rate = 0.02/252  # Assuming daily risk-free rate for simplicity dividing by 25
avg_daily_return = equity_returns.mean()
std_daily_return = equity_returns.std()
excess_return = avg_daily_return - risk_free_rate
sharpe_ratio = excess_return / std_daily_return
annualized_sharpe_ratio=sharpe_ratio*np.sqrt(252) #annualize it
for i, equity in enumerate(equities):
    print(f"Annualized Sharpe Ratio for {equity}: {annualized_sharpe_ratio[i]}")
```

```
Annualized Sharpe Ratio for AAPL: 0.9308329009910372
Annualized Sharpe Ratio for MSFT: 0.7299861814778429
Annualized Sharpe Ratio for AMZN: 0.5924806931157081
Annualized Sharpe Ratio for GOOGL: 0.5522289340991853
Annualized Sharpe Ratio for JNJ: 0.7922442277499284
```

In [57]:

```
risk_free_rate=0.02/252
avg_daily_return=index_returns.mean()
std_daily_return=index_returns.std()
excess_return=avg_daily_return-risk_free_rate
sharpe_ratio=excess_return/std_daily_return
annualized_sharpe_ratio=sharpe_ratio*np.sqrt(252)

for i, index in enumerate(indices) :
    print(f"Annualized Sharpe Ratio for {index}: {annualized_sharpe_ratio[i]}")
```

```
Annualized Sharpe Ratio for ^GSPC: 0.4726315135289208
Annualized Sharpe Ratio for ^DJI: 0.11952124472370917
Annualized Sharpe Ratio for ^IXIC: 0.3542481582494824
Annualized Sharpe Ratio for ^FTSE: 0.5184290256747393
Annualized Sharpe Ratio for ^GDAXI: 0.604267311328777
```

In [58]:

```
# calculation of sortino ratio for equity

risk_free_rate=0.02/252
downside_returns= equity_returns[equity_returns<0]
downside_std_deviation=downside_returns.std()

avg_daily_returns=equity_returns.mean()
excess_return= avg_daily_returns-risk_free_rate

sortino_ratio=excess_return/downside_std_deviation
annualized_sortino_ratio=sortino_ratio*np.sqrt(252)

for i, equity in enumerate(equities):
    print(f"Annualized Sortino Ratio for {equity}: {annualized_sortino_ratio[i]}")
```

```
Annualized Sortino Ratio for AAPL: 1.2999856246062589
Annualized Sortino Ratio for MSFT: 1.0424225429595226
Annualized Sortino Ratio for AMZN: 0.8276058865492264
Annualized Sortino Ratio for GOOGL: 0.7322967742665923
Annualized Sortino Ratio for JNJ: 1.1075161908041606
```

In [59]:

```
# calculation of sortino ratio for index

risk_free_rate=0.02/252
downside_returns= index_returns[index_returns<0]
downside_std_deviation=downside_returns.std()

avg_daily_returns=index_returns.mean()
excess_return= avg_daily_returns-risk_free_rate

sortino_ratio=excess_return/downside_std_deviation
annualized_sortino_ratio=sortino_ratio*np.sqrt(252)

for i, index in enumerate(indices):
    print(f"Annualized Sortino Ratio for {index}: {annualized_sortino_ratio[i]}")
```

```
Annualized Sortino Ratio for ^GSPC: 0.5638427507893821
Annualized Sortino Ratio for ^DJI: 0.15260686314463975
Annualized Sortino Ratio for ^IXIC: 0.4602473278967329
Annualized Sortino Ratio for ^FTSE: 0.6265258924722723
Annualized Sortino Ratio for ^GDAXI: 0.7533371637749335
```

In [ ]:

```

```