In [13]:
```
!pip install yfinance pandas numpy matplotlib --quiet
```

In [14]:
```
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [19]:
```
# Step 3: Download only 'Close' prices from Yahoo Finance
tickers = ['AAPL', 'MSFT', 'GOOGL', 'AMZN']

# Download data and filter only 'Close'
data = yf.download(tickers, start='2020-01-01', end='2024-01-01', auto_adjust=Tr

# Remove multi-level column (only ticker names remain)
data.columns = data.columns.droplevel(0)

# View clean data
data.head()
```

```
[**********************100%**********************]  4 of 4 completed
```

Out[19]:

Ticker	AAPL	AMZN	GOOGL	MSFT
Date				
2020-01-02	72.620834	94.900497	68.026031	153.042297
2020-01-03	71.914810	93.748497	67.670143	151.136673
2020-01-06	72.487854	95.143997	69.473846	151.527298
2020-01-07	72.146935	95.343002	69.339645	150.145721
2020-01-08	73.307518	94.598503	69.833183	152.537277

In [20]:
```
# Step 4: Calculate Daily Returns
returns = data.pct_change().dropna()
returns.head()
```

Out[20]:

Ticker	AAPL	AMZN	GOOGL	MSFT
Date				
2020-01-03	-0.009722	-0.012139	-0.005232	-0.012452
2020-01-06	0.007968	0.014886	0.026654	0.002585
2020-01-07	-0.004703	0.002092	-0.001932	-0.009118
2020-01-08	0.016086	-0.007809	0.007118	0.015928
2020-01-09	0.021241	0.004799	0.010498	0.012493

In [21]:
```
weights = np.array([0.25, 0.25, 0.25, 0.25])
print("✅ Portfolio Weights:", weights)
```

```
✅ Portfolio Weights: [0.25 0.25 0.25 0.25]
```

In [22]:
```
mean_returns = returns.mean() * 252
expected_return = np.dot(weights, mean_returns)
print("✅ Expected Annual Portfolio Return: {:.2f}%".format(expected_return * 1
```

```
✅ Expected Annual Portfolio Return: 24.98%
```

In [23]:
```
# Annualized Covariance Matrix of Returns
cov_matrix = returns.cov() * 252

# Portfolio Standard Deviation (Volatility / Risk)
portfolio_std = np.sqrt(np.dot(weights.T, np.dot(cov_matrix, weights)))

print("⚠️ Portfolio Risk (Standard Deviation): {:.2f}%".format(portfolio_std * 1
```

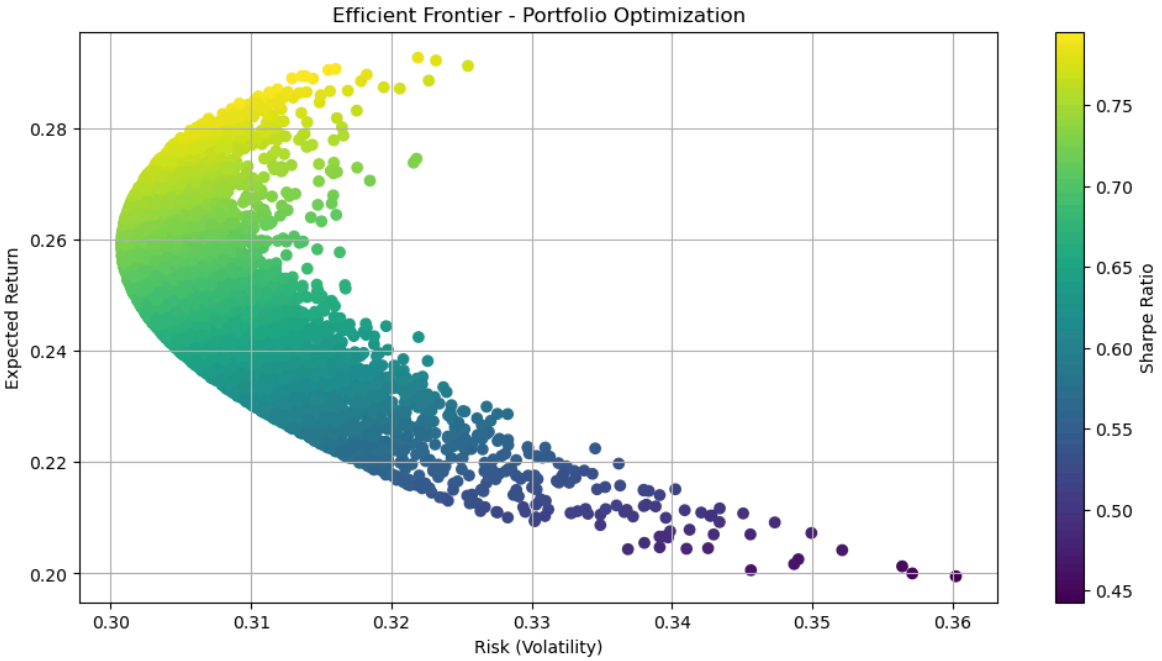
```
⚠️ Portfolio Risk (Standard Deviation): 30.22%
```

In [24]:
```
# Assume a constant risk-free rate (e.g., 4% for fixed deposits)
risk_free_rate = 0.04

# Sharpe Ratio Formula
sharpe_ratio = (expected_return - risk_free_rate) / portfolio_std

print("📊 Sharpe Ratio: {:.2f}".format(sharpe_ratio))
```

```
📊 Sharpe Ratio: 0.69
```

In [26]:
```
num_portfolios = 10000
all_returns = []
all_volatilities = []
all_sharpes = []
all_weights = []

for _ in range(num_portfolios):
    # Random weights that sum to 1
    w = np.random.random(len(tickers))
    w /= np.sum(w)

    # Expected return for this set of weights
    ret = np.dot(w, mean_returns)

    # Portfolio risk
    vol = np.sqrt(np.dot(w.T, np.dot(cov_matrix, w)))

    # Sharpe ratio
    sharpe = (ret - risk_free_rate) / vol

    # Append results
    all_returns.append(ret)
    all_volatilities.append(vol)
    all_sharpes.append(sharpe)
    all_weights.append(w)
```

In [27]:
```
plt.figure(figsize=(12, 6))
scatter = plt.scatter(all_volatilities, all_returns, c=all_sharpes, cmap='viridi
plt.colorbar(scatter, label='Sharpe Ratio')
plt.xlabel('Risk (Volatility)')
plt.ylabel('Expected Return')
plt.title('Efficient Frontier - Portfolio Optimization')
plt.grid(True)
plt.show()
```



In [28]:
```python
# Convert Sharpe ratios to a NumPy array
all_sharpes = np.array(all_sharpes)

# Find the index of the portfolio with the highest Sharpe Ratio
max_sharpe_idx = all_sharpes.argmax()

# Get details of the optimal portfolio
optimal_return = all_returns[max_sharpe_idx]
optimal_volatility = all_volatilities[max_sharpe_idx]
optimal_weights = all_weights[max_sharpe_idx]

print("🏆 Optimal Portfolio (Highest Sharpe Ratio):")
for i, ticker in enumerate(tickers):
    print(f"{ticker}: {optimal_weights[i]*100:.2f}%")

print(f"\n✅ Expected Return: {optimal_return*100:.2f}%")
print(f"⚠️ Volatility (Risk): {optimal_volatility*100:.2f}%")
print(f"📊 Sharpe Ratio: {all_sharpes[max_sharpe_idx]:.2f}")
```

```
🏆 Optimal Portfolio (Highest Sharpe Ratio):
AAPL: 56.75%
MSFT: 0.02%
GOOGL: 0.06%
AMZN: 43.17%

✅ Expected Return: 28.89%
⚠️ Volatility (Risk): 31.29%
📊 Sharpe Ratio: 0.80
```

In [ ]: