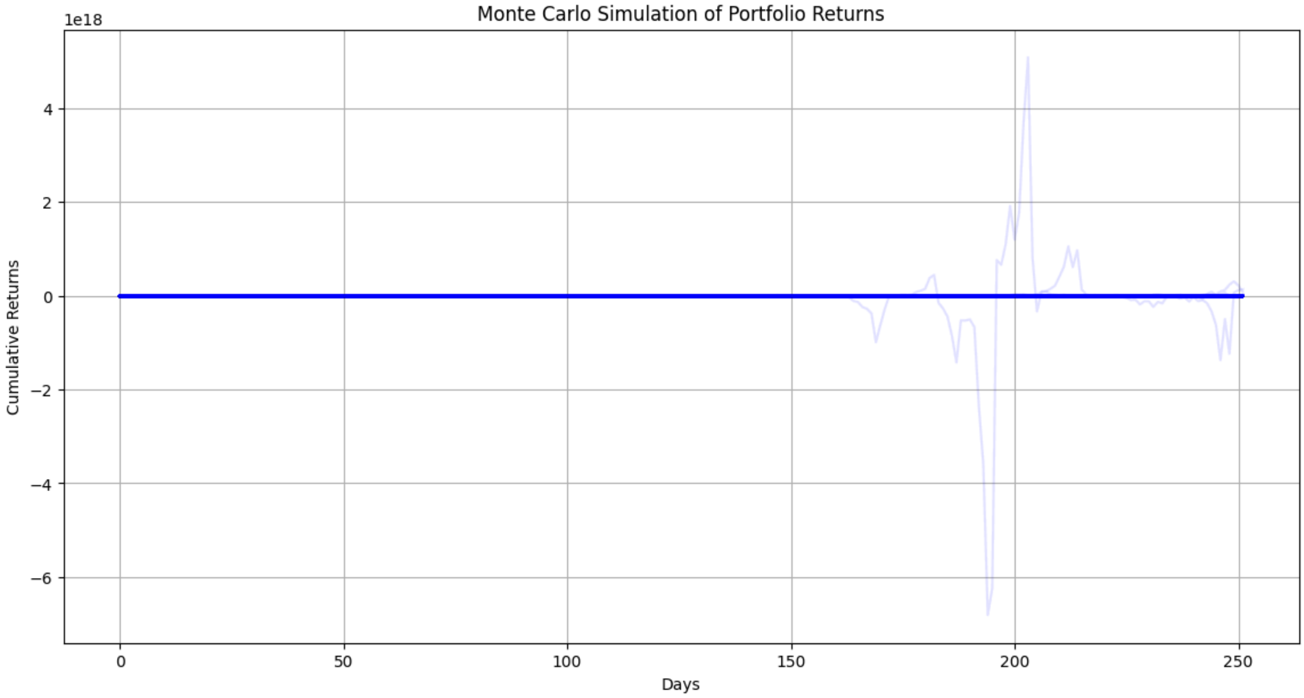
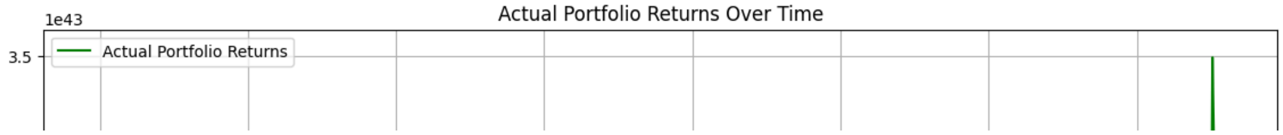
```
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import timedelta

nifty_50_stocks = [
    'ADANIPORTS.NS', 'ASIANPAINT.NS', 'AXISBANK.NS', 'BAJAJ-AUTO.NS',
    'BAJAJFINANCE.NS', 'BAJAJFINSV.NS', 'BPCL.NS', 'BHARTIARTL.NS',
    'BRITANNIA.NS', 'CIPLA.NS', 'COALINDIA.NS',
    'DRREDDY.NS', 'EICHERMOT.NS', 'GAIL.NS', 'GRASIM.NS',
    'HCLTECH.NS', 'HDFCBANK.NS', 'HEROMOTOCO.NS', 'HINDALCO.NS',
    'HINDUNILVR.NS', 'ITC.NS', 'ICICIBANK.NS',
    'IOC.NS', 'INDUSINDBK.NS', 'INFY.NS',
    'JSWSTEEL.NS', 'KOTAKBANK.NS', 'LT.NS', 'M&M.NS',
    'MARUTI.NS', 'NTPC.NS', 'NESTLEIND.NS', 'ONGC.NS',
    'POWERGRID.NS', 'RELIANCE.NS', 'SBIN.NS', 'SUNPHARMA.NS',
    'TCS.NS', 'TATAMOTORS.NS', 'TATASTEEL.NS', 'TECHM.NS',
    'TITAN.NS', 'UPL.NS', 'ULTRACEMCO.NS', 'VEDL.NS',
    'WIPRO.NS', 'YESBANK.NS'
]

short_window = 50
long_window = 200
start_date = '2000-01-01'
end_date = '2024-01-01'

risk_free_rate = 0.07
stop_loss_percentage = 0.08

trade_order_book = []

def generate_signals_with_sma(stock_data):
    stock_data['SMA50'] = stock_data['Close'].rolling(window=short_window, min_periods=1).mean()
    stock_data['SMA200'] = stock_data['Close'].rolling(window=long_window, min_periods=1).mean()

    stock_data['in_trade'] = 0
    stock_data['Signal'] = np.where((stock_data['in_trade'] == 0) & (stock_data['SMA50'] > stock_data['SMA200']), 1, 0)
    stock_data['Signal'] = np.where((stock_data['in_trade'] == 1) & (stock_data['SMA50'] < stock_data['SMA200']), -1, stock_data['Signa
    stock_data['Position'] = stock_data['Signal'].diff()
    stock_data['in_trade'] = np.where(stock_data['Position'] == 1, 1,
                                      np.where(stock_data['Position'] == -1, 0, stock_data['in_trade']))

    stock_data['Returns'] = stock_data['Close'].pct_change()
    return stock_data


for stock_symbol in nifty_50_stocks:
    print(f"Processing {stock_symbol}")
    stock_data = yf.download(stock_symbol, start=start_date, end=end_date)
    stock_data = generate_signals_with_sma(stock_data)

    for i in range(len(stock_data)):
        if stock_data['Position'].iloc[i] == 1:
            buy_price = stock_data['Close'].iloc[i]
            buy_date = stock_data.index[i]
            quantity = 1
            trade_order_book.append([stock_symbol, buy_price, None, buy_date, None, None, quantity])
        elif stock_data['Position'].iloc[i] == -1:
            for trade in reversed(trade_order_book):
                if trade[0] == stock_symbol and trade[2] is None:
                    sell_price = stock_data['Close'].iloc[i]
                    sell_date = stock_data.index[i]
                    profit_loss = sell_price - trade[1]
                    trade[2] = sell_price
                    trade[4] = sell_date
                    trade[5] = profit_loss
                    break

final_date = pd.to_datetime("2024-01-01")
for trade in trade_order_book:
    if trade[2] is None:
        final_price = yf.download(trade[0], start=final_date, end=final_date + timedelta(days=1))['Close'].values[0]
        trade[2] = final_price
        trade[4] = final_date
        trade[5] = final_price - trade[1]

trade_order_df = pd.DataFrame(trade_order_book, columns=['Ticker', 'Buy Price', 'Sell Price', 'Buy Date', 'Sell Date', 'Profit/Loss', '
trade_order_df['Stop Loss'] = trade_order_df['Buy Price'] * (1 - stop_loss_percentage)
trade_order_df['Total Profit/Loss'] = trade_order_df['Profit/Loss'] * trade_order_df['Quantity']
```

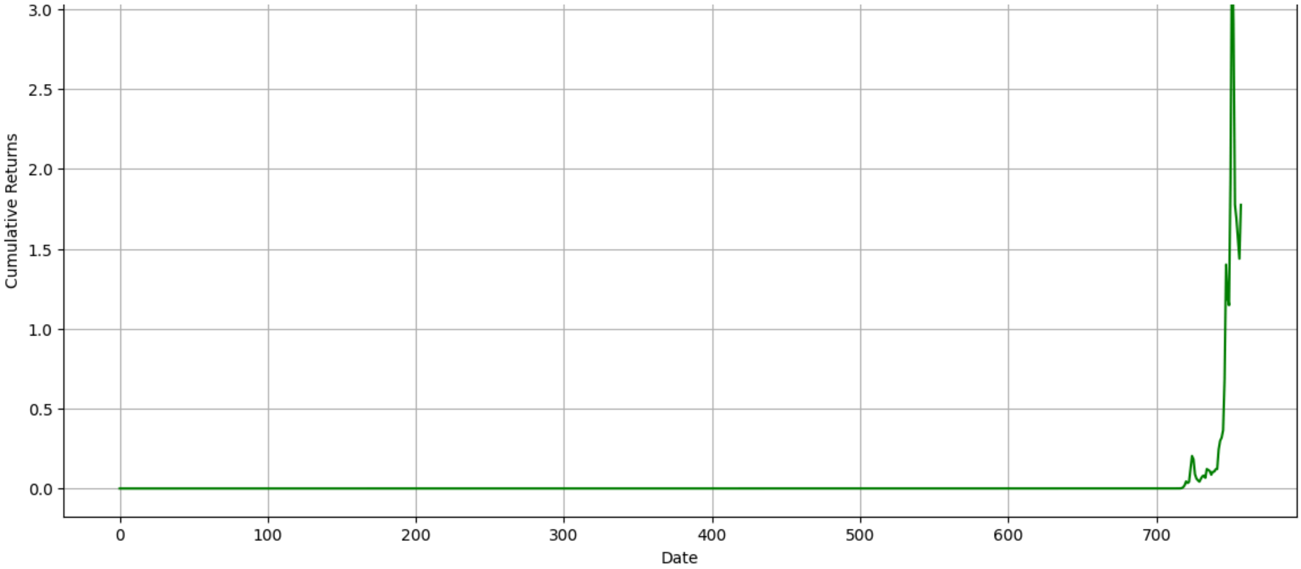
```
if not trade_order_df.empty:
    trade_order_df['Returns'] = trade_order_df['Profit/Loss'] / trade_order_df['Buy Price']
    total_return = trade_order_df['Profit/Loss'].sum()
    portfolio_returns = (1 + trade_order_df['Returns']).cumprod()

    years = (trade_order_df['Sell Date'].max() - trade_order_df['Buy Date'].min()).days / 365
    cagr = (portfolio_returns.iloc[-1]) ** (1 / years) - 1
    std_dev = trade_order_df['Profit/Loss'].std() * np.sqrt(252)
    sharpe_ratio = (portfolio_returns.mean() * 252 - risk_free_rate) / std_dev if std_dev != 0 else np.nan
    roll_max = portfolio_returns.cummax()
    drawdown = portfolio_returns / roll_max - 1
    max_drawdown = drawdown.min()

    negative_returns = trade_order_df['Returns'][trade_order_df['Returns'] < 0]
    downside_deviation = np.std(negative_returns) * np.sqrt(252)
    sortino_ratio = (portfolio_returns.mean() * 252 - risk_free_rate) / downside_deviation if downside_deviation != 0 else np.nan

    print(f'Total Portfolio Return: {total_return}')
    print(f'CAGR: {cagr}')
    print(f'Standard Deviation: {std_dev}')
    print(f'Sharpe Ratio: {sharpe_ratio}')
    print(f'Maximum Drawdown: {max_drawdown}')
    print(f'Sortino Ratio: {sortino_ratio}')

    num_simulations = 1000
    num_days = 252
    simulation_results = np.zeros((num_simulations, num_days))

    mean_return = trade_order_df['Returns'].mean()
    std_return = trade_order_df['Returns'].std()

    for i in range(num_simulations):
        daily_returns = np.random.normal(mean_return, std_return, num_days)
        cumulative_returns = np.cumprod(1 + daily_returns)
        simulation_results[i] = cumulative_returns

    plt.figure(figsize=(14, 7))
    plt.plot(simulation_results.T, color='blue', alpha=0.1)
    plt.title('Monte Carlo Simulation of Portfolio Returns')
    plt.xlabel('Days')
    plt.ylabel('Cumulative Returns')
    plt.grid()
    plt.show()

    plt.figure(figsize=(14, 7))
    plt.plot(portfolio_returns, color='green', label='Actual Portfolio Returns')
    plt.title('Actual Portfolio Returns Over Time')
    plt.xlabel('Date')
    plt.ylabel('Cumulative Returns')
    plt.grid()
    plt.legend()
    plt.show()

else:
    print("No trades were executed. Portfolio metrics cannot be calculated.")
```

```
Processing ADANIPORTS.NS
[*********************100%***********************]  1 of 1 completed
Processing ASIANPAINT.NS
[*********************100%***********************]  1 of 1 completed
Processing AXISBANK.NS
[*********************100%***********************]  1 of 1 completed
Processing BAJAJ-AUTO.NS
[*********************100%***********************]  1 of 1 completed
Processing BAJAJFINANCE.NS
[*********************100%***********************]  1 of 1 completed
ERROR:yfinance:
1 Failed download:
ERROR:yfinance:['BAJAJFINANCE.NS']: YFTzMissingError('$%ticker%: possibly delisted; no timezone found')
Processing BAJAJFINSV.NS
[*********************100%***********************]  1 of 1 completed
Processing BPCL.NS
[*********************100%***********************]  1 of 1 completed
Processing BHARTIARTL.NS
[*********************100%***********************]  1 of 1 completed
Processing BRITANNIA.NS
[*********************100%***********************]  1 of 1 completed
Processing CIPLA.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completedProcessing COALINDIA.NS

Processing DRREDDY.NS
[*********************100%***********************]  1 of 1 completed
Processing EICHERMOT.NS
[*********************100%***********************]  1 of 1 completed
Processing GAIL.NS
[*********************100%***********************]  1 of 1 completed
Processing GRASIM.NS
[*********************100%***********************]  1 of 1 completed
Processing HCLTECH.NS
[*********************100%***********************]  1 of 1 completed
Processing HDFCBANK.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completedProcessing HEROMOTOCO.NS

Processing HINDALCO.NS
[*********************100%***********************]  1 of 1 completed
Processing HINDUNILVR.NS
[*********************100%***********************]  1 of 1 completed
Processing ITC.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completedProcessing ICICIBANK.NS

Processing IOC.NS
[*********************100%***********************]  1 of 1 completed
Processing INDUSINDBK.NS
[*********************100%***********************]  1 of 1 completed
Processing INFY.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completedProcessing JSWSTEEL.NS

Processing KOTAKBANK.NS
[*********************100%***********************]  1 of 1 completed
Processing LT.NS
[*********************100%***********************]  1 of 1 completed
Processing M&M.NS
[*********************100%***********************]  1 of 1 completed
Processing MARUTI.NS
[*********************100%***********************]  1 of 1 completed
Processing NTPC.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completedProcessing NESTLEIND.NS

Processing ONGC.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completedProcessing POWERGRID.NS

Processing RELIANCE.NS
[*********************100%***********************]  1 of 1 completed
Processing SBIN.NS
[*********************100%***********************]  1 of 1 completed
Processing SUNPHARMA.NS
[*********************100%***********************]  1 of 1 completed
Processing TCS.NS
[*********************100%***********************]  1 of 1 completed
Processing TATAMOTORS.NS
[*********************100%***********************]  1 of 1 completed
Processing TATASTEEL.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completedProcessing TECHM.NS

Processing TITAN.NS
[*********************100%***********************]  1 of 1 completed
Processing UPL.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completedProcessing ULTRACEMCO.NS
```

```
Processing VEDL.NS
[*********************100%***********************]  1 of 1 completed
Processing WIPRO.NS
[*********************100%***********************]  1 of 1 completed
Processing YESBANK.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
Total Portfolio Return: 54972.699986457825
CAGR: 64.42249564969637
Standard Deviation: 5946.991703132816
Sharpe Ratio: 1.3589873066109644e+40
Maximum Drawdown: -0.9238040512469411
Sortino Ratio: 4.2032137995745234e+43
```






3.0
2.5
2.0
1.5
1.0
0.5
0.0
Cumulative Returns
0
100
200
300
400
500
600
700
Date

```
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import timedelta
from google.colab import files

nifty_50_stocks = [
    'ADANIPORTS.NS', 'ASIANPAINT.NS', 'AXISBANK.NS', 'BAJAJ-AUTO.NS',
    'BAJAJFINANCE.NS', 'BAJAJFINSV.NS', 'BPCL.NS', 'BHARTIARTL.NS',
    'BRITANNIA.NS', 'CIPLA.NS', 'COALINDIA.NS',
    'DRREDDY.NS', 'EICHERMOT.NS', 'GAIL.NS', 'GRASIM.NS',
    'HCLTECH.NS', 'HDFCBANK.NS', 'HEROMOTOCO.NS', 'HINDALCO.NS',
    'HINDUNILVR.NS', 'ITC.NS', 'ICICIBANK.NS',
    'IOC.NS', 'INDUSINDBK.NS', 'INFY.NS',
    'JSWSTEEL.NS', 'KOTAKBANK.NS', 'LT.NS', 'M&M.NS',
    'MARUTI.NS', 'NTPC.NS', 'NESTLEIND.NS', 'ONGC.NS',
    'POWERGRID.NS', 'RELIANCE.NS', 'SBIN.NS', 'SUNPHARMA.NS',
    'TCS.NS', 'TATAMOTORS.NS', 'TATASTEEL.NS', 'TECHM.NS',
    'TITAN.NS', 'UPL.NS', 'ULTRACEMCO.NS', 'VEDL.NS',
    'WIPRO.NS', 'YESBANK.NS'
]

short_window = 50
long_window = 200
start_date = '2000-01-01'
end_date = '2024-01-01'

risk_free_rate = 0.07
stop_loss_percentage = 0.08

trade_order_book = []

def generate_signals_with_sma(stock_data):
    stock_data['SMA50'] = stock_data['Close'].rolling(window=short_window, min_periods=1).mean()
    stock_data['SMA200'] = stock_data['Close'].rolling(window=long_window, min_periods=1).mean()

    stock_data['in_trade'] = 0

    stock_data['Signal'] = np.where((stock_data['in_trade'] == 0) & (stock_data['SMA50'] > stock_data['SMA200']), 1, 0)

    stock_data['Signal'] = np.where((stock_data['in_trade'] == 1) & (stock_data['SMA50'] < stock_data['SMA200']), -1, stock_data['Signa

    stock_data['Position'] = stock_data['Signal'].diff()

    stock_data['in_trade'] = np.where(stock_data['Position'] == 1, 1,
                                       np.where(stock_data['Position'] == -1, 0, stock_data['in_trade']))

    stock_data['Returns'] = stock_data['Close'].pct_change()

    return stock_data

for stock_symbol in nifty_50_stocks:
    print(f"Processing {stock_symbol}")
    stock_data = yf.download(stock_symbol, start=start_date, end=end_date)
    stock_data = generate_signals_with_sma(stock_data)

    for i in range(len(stock_data)):
        if stock_data['Position'].iloc[i] == 1:
            buy_price = stock_data['Close'].iloc[i]
            buy_date = stock_data.index[i]
            quantity = 1
            trade_order_book.append([stock_symbol, buy_price, None, buy_date, None, None, quantity])
        elif stock_data['Position'].iloc[i] == -1:
            for trade in reversed(trade_order_book):
                if trade[0] == stock_symbol and trade[2] is None:
                    sell_price = stock_data['Close'].iloc[i]
                    sell_date = stock_data.index[i]
                    profit_loss = sell_price - trade[1]
                    trade[2] = sell_price
                    trade[4] = sell_date
                    trade[5] = profit_loss
                    break

final_date = pd.to_datetime("2024-01-01")
for trade in trade_order_book:
    if trade[2] is None:
        final_price = yf.download(trade[0], start=final_date, end=final_date + timedelta(days=1))['Close'].values[0]
        trade[2] = final_price
        trade[4] = final_date
        trade[5] = final_price - trade[1]
```

```
trade_order_df = pd.DataFrame(trade_order_book, columns=['Ticker', 'Buy Price', 'Sell Price', 'Buy Date', 'Sell Date', 'Profit/Loss', '(

trade_order_df['Stop Loss'] = trade_order_df['Buy Price'] * (1 - stop_loss_percentage)
trade_order_df['Total Profit/Loss'] = trade_order_df['Profit/Loss'] * trade_order_df['Quantity']


trade_order_df.to_excel('trade_order_book.xlsx', index=False)

files.download('trade_order_book.xlsx')

if not trade_order_df.empty:
    trade_order_df['Returns'] = trade_order_df['Profit/Loss'] / trade_order_df['Buy Price']
    total_return = trade_order_df['Profit/Loss'].sum()
    portfolio_returns = (1 + trade_order_df['Returns']).cumprod()

    years = (trade_order_df['Sell Date'].max() - trade_order_df['Buy Date'].min()).days / 365
    cagr = (portfolio_returns.iloc[-1]) ** (1 / years) - 1
    std_dev = trade_order_df['Profit/Loss'].std() * np.sqrt(252)
    sharpe_ratio = (portfolio_returns.mean() * 252 - risk_free_rate) / std_dev if std_dev != 0 else np.nan
    roll_max = portfolio_returns.cummax()
    drawdown = portfolio_returns / roll_max - 1
    max_drawdown = drawdown.min()

    negative_returns = trade_order_df['Returns'][trade_order_df['Returns'] < 0]
    downside_deviation = np.std(negative_returns) * np.sqrt(252)
    sortino_ratio = (portfolio_returns.mean() * 252 - risk_free_rate) / downside_deviation if downside_deviation != 0 else np.nan

    print(f'Total Portfolio Return: {total_return}')
    print(f'CAGR: {cagr}')
    print(f'Standard Deviation: {std_dev}')
    print(f'Sharpe Ratio: {sharpe_ratio}')
    print(f'Maximum Drawdown: {max_drawdown}')
    print(f'Sortino Ratio: {sortino_ratio}')
else:
    print("No trades were executed. Portfolio metrics cannot be calculated.")

trade_order_df
```

```
[*********************100%***********************]  1 of 1 completedProcessing ADANIPORTS.NS
Processing ASIANPAINT.NS

[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
Processing AXISBANK.NS
[*********************100%***********************]  1 of 1 completedProcessing BAJAJ-AUTO.NS

Processing BAJAJFINANCE.NS
[*********************100%***********************]  1 of 1 completed
ERROR:yfinance:
1 Failed download:
ERROR:yfinance:['BAJAJFINANCE.NS']: YFTzMissingError('$%ticker%: possibly delisted; no timezone found')
[*********************100%***********************]  1 of 1 completed
Processing BAJAJFINSV.NS
Processing BPCL.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
Processing BHARTIARTL.NS
Processing BRITANNIA.NS
[*********************100%***********************]  1 of 1 completed
Processing CIPLA.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completedProcessing COALINDIA.NS
Processing DRREDDY.NS

[*********************100%***********************]  1 of 1 completed
Processing EICHERMOT.NS
[*********************100%***********************]  1 of 1 completedProcessing GAIL.NS

[*********************100%***********************]  1 of 1 completed
Processing GRASIM.NS
Processing HCLTECH.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
Processing HDFCBANK.NS
[*********************100%***********************]  1 of 1 completed
Processing HEROMOTOCO.NS
Processing HINDALCO.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completedProcessing HINDUNILVR.NS

[*********************100%***********************]  1 of 1 completedProcessing ITC.NS

[*********************100%***********************]  1 of 1 completedProcessing ICICIBANK.NS
Processing IOC.NS

[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completedProcessing INDUSINDBK.NS

[*********************100%***********************]  1 of 1 completed
Processing INFY.NS
[*********************100%***********************]  1 of 1 completed
Processing JSWSTEEL.NS
Processing KOTAKBANK.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
Processing LT.NS
Processing M&M.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
Processing MARUTI.NS
Processing NTPC.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
Processing NESTLEIND.NS
Processing ONGC.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
Processing POWERGRID.NS
Processing RELIANCE.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completedProcessing SBIN.NS

[*********************100%***********************]  1 of 1 completedProcessing SUNPHARMA.NS

[*********************100%***********************]  1 of 1 completed
Processing TCS.NS
Processing TATAMOTORS.NS
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completedProcessing TATASTEEL.NS

[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completedProcessing TECHM.NS
Processing TITAN.NS

[*********************100%***********************]  1 of 1 completedProcessing UPL.NS

[*********************100%***********************]  1 of 1 completedProcessing ULTRACEMCO.NS
```

```
[*********************100%***********************]  1 of 1 completedProcessing VEDL.NS

[*********************100%***********************]  1 of 1 completedProcessing WIPRO.NS

[*********************100%***********************]  1 of 1 completedProcessing YESBANK.NS

[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
Total Portfolio Return: 54972.699986457825
CAGR: 64.42249564969637
Standard Deviation: 5946.991703132816
Sharpe Ratio: 1.3589873066109644e+40
Maximum Drawdown: -0.9238040512469411
Sortino Ratio: 4.2032137995745234e+43
```

1 to 25 of 758 entries Filter ?

index	Ticker	Buy Price	Sell Price	Buy Date	Sell Date	Profit/Loss	Quantity	Stop Loss	Total Pro
0	ADANIPORTS.NS	159.25	152.77999877929688	2008-02-07 00:00:00	2008-02-19 00:00:00	-6.470001220703125	1	146.51000000000002	-6.4700012
1	ADANIPORTS.NS	122.23999786376953	144.0500030517578	2009-05-29 00:00:00	2010-12-31 00:00:00	21.81000518798828	1	112.46079803466797	21.810005
2	ADANIPORTS.NS	162.35000610351562	129.0	2011-07-01 00:00:00	2011-12-12 00:00:00	-33.350006103515625	1	149.36200561523438	-33.350006
3	ADANIPORTS.NS	132.25	123.5	2012-12-03 00:00:00	2013-08-27 00:00:00	-8.75	1	121.67	
4	ADANIPORTS.NS	152.0500030517578	291.20001220703125	2013-11-21 00:00:00	2015-11-03 00:00:00	139.15000915527344	1	139.8860028076172	139.15000
5	ADANIPORTS.NS	257.70001220703125	379.29998779296875	2016-08-26 00:00:00	2018-04-06 00:00:00	121.5999755859375	1	237.08401123046875	121.5999
6	ADANIPORTS.NS	396.20001220703125	339.54998779296875	2019-01-21 00:00:00	2019-02-19 00:00:00	-56.6500244140625	1	364.5040112304688	-56.6500
7	ADANIPORTS.NS	354.75	324.54998779296875	2019-02-20 00:00:00	2019-02-25 00:00:00	-30.20001220703125	1	326.37	-30.200012
8	ADANIPORTS.NS	391.75	361.8999938964844	2019-04-26 00:00:00	2019-09-19 00:00:00	-29.850006103515625	1	360.41	-29.850006
9	ADANIPORTS.NS	421.70001220703125	369.6499938964844	2019-10-18 00:00:00	2019-12-19 00:00:00	-52.050018310546875	1	387.96401123046877	-52.050018
10	ADANIPORTS.NS	360.04998779296875	722.75	2020-08-28	2021-12-29	362.70001220703125	1	331.24598876953127	362.700012

				00:00:00	00:00:00				
11	ADANIPORTS.NS	762.9500122070312	702.4500122070312	2022-01-18 00:00:00	2022-01-24 00:00:00	-60.5	1	701.9140112304688	
12	ADANIPORTS.NS	745.2000122070312	732.5	2022-02-02 00:00:00	2022-02-03 00:00:00	-12.70001220703125	1	685.5840112304688	-12.700012
13	ADANIPORTS.NS	733.4500122070312	695.75	2022-02-09 00:00:00	2022-02-14 00:00:00	-37.70001220703125	1	674.7740112304688	-37.700012
14	ADANIPORTS.NS	720.5999755859375	740.75	2022-02-15 00:00:00	2022-02-16 00:00:00	20.1500244140625	1	662.9519775390626	20.15002
15	ADANIPORTS.NS	849.8499755859375	682.0	2022-04-06 00:00:00	2022-07-04 00:00:00	-167.8499755859375	1	781.8619775390625	-167.84997
16	ADANIPORTS.NS	837.7000122070312	545.4500122070312	2022-08-26 00:00:00	2023-02-06 00:00:00	-292.25	1	770.6840112304687	
17	ADANIPORTS.NS	765.3499755859375	1047.8499755859375	2023-08-01 00:00:00	2024-01-01 00:00:00	282.5	1	704.1219775390625	
18	ASIANPAINT.NS	22.906999588012695	22.966999053955078	2002-09-09 00:00:00	2002-09-13 00:00:00	0.05999946594238281	1	21.07443962097168	0.059999465
19	ASIANPAINT.NS	22.68000030517578	21.32699966430664	2002-10-18 00:00:00	2002-11-12 00:00:00	-1.3530006408691406	1	20.86560028076172	-1.35300064
20	ASIANPAINT.NS	24.683000564575195	30.55500030517578	2003-05-16 00:00:00	2004-05-03 00:00:00	5.871999740600586	1	22.70836051940918	5.8719997
21	ASIANPAINT.NS	31.219999313354492	29.53499984741211	2004-08-23 00:00:00	2004-10-29 00:00:00	-1.6849994659423828	1	28.722399368286133	-1.68499946
22	ASIANPAINT.NS	30.94499969482422	57.005001068115234	2005-01-06 00:00:00	2006-07-12 00:00:00	26.060001373291016	1	28.469399719238282	26.0600013
23	ASIANPAINT.NS	63.79999923706055	120.30999755859375	2006-09-01 00:00:00	2008-09-05 00:00:00	56.5099983215332	1	58.69599929809571	56.50999
				2008-09-	2008-10-				

```
!pip install backtrader
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import timedelta
import backtrader as bt

nifty_50_stocks = [
    'ADANIPORTS.NS', 'ASIANPAINT.NS', 'AXISBANK.NS', 'BAJAJ-AUTO.NS',
    'BAJAJFINSV.NS', 'BPCL.NS', 'BHARTIARTL.NS',
    'BRITANNIA.NS', 'CIPLA.NS', 'COALINDIA.NS', 'DRREDDY.NS',
    'EICHERMOT.NS', 'GAIL.NS', 'GRASIM.NS', 'HCLTECH.NS',
    'HDFCBANK.NS', 'HEROMOTOCO.NS', 'HINDALCO.NS', 'HINDUNILVR.NS',
    'ITC.NS', 'ICICIBANK.NS', 'IOC.NS', 'INDUSINDBK.NS',
    'INFY.NS', 'JSWSTEEL.NS', 'KOTAKBANK.NS', 'LT.NS',
    'M&M.NS', 'MARUTI.NS', 'NTPC.NS', 'NESTLEIND.NS',
    'ONGC.NS', 'POWERGRID.NS', 'RELIANCE.NS', 'SBIN.NS',
    'SUNPHARMA.NS', 'TCS.NS', 'TATAMOTORS.NS', 'TATASTEEL.NS',
    'TECHM.NS', 'TITAN.NS', 'UPL.NS', 'ULTRACEMCO.NS',
    'VEDL.NS', 'WIPRO.NS', 'YESBANK.NS'
]

class SMACross(bt.Strategy):
    params = (("short_window", 50), ("long_window", 200),)

    def __init__(self):
        self.sma_short = bt.indicators.SimpleMovingAverage(
            self.data.close, period=self.params.short_window
        )
        self.sma_long = bt.indicators.SimpleMovingAverage(
            self.data.close, period=self.params.long_window
        )
        self.order = None

    def next(self):
        if self.order:
            return

        if self.sma_short > self.sma_long and not self.position:
            self.order = self.buy()

        elif self.sma_short < self.sma_long and self.position:
            self.order = self.sell()

def run_backtest(stock_symbol):
    print(f"Running backtest for {stock_symbol}")

    cerebro = bt.Cerebro()
    cerebro.addstrategy(SMACross)

    stock_data = yf.download(stock_symbol, start='2000-01-01', end='2024-01-01')

    stock_data.index = pd.to_datetime(stock_data.index)

    data = bt.feeds.PandasData(dataname=stock_data)

    cerebro.adddata(data)

    cerebro.broker.setcash(100000)
    cerebro.run()

    portfolio_value = cerebro.broker.getvalue()
    print(f"Final Portfolio Value: {portfolio_value}")

    cerebro.plot()

for stock_symbol in nifty_50_stocks:
    run_backtest(stock_symbol)
```

```
Requirement already satisfied: backtrader in /usr/local/lib/python3.10/dist-packages (1.9.78.123)
Running backtest for ADANIPORTS.NS
[*********************100%***********************]  1 of 1 completed
Final Portfolio Value: 100899.44997406006
[*********************100%***********************]  1 of 1 completedRunning backtest for ASIANPAINT.NS
```