Bot

```
# Install required libraries
# pip install streamlit pandas numpy ta yfinance

import streamlit as st
import pandas as pd
import numpy as np
import ta  # For technical indicators
import yfinance as yf  # For fetching live market data

# Streamlit App Configuration
st.title("Trading Signal Dashboard")
st.sidebar.header("Settings")

# User Inputs
ticker = st.sidebar.text_input("Enter Stock/Asset Ticker (e.g., AAPL, BTC-USD)", value="AAPL")
timeframe = st.sidebar.selectbox("Select Timeframe", ["1m", "5m", "15m", "1h", "1d"])
```

```
indicator = st.sidebar.selectbox("Select Indicator", ["SMA", "EMA", "RSI", "MACD", "Bollinger Bands"])

# Fetch Live Data
@st.cache
def fetch_data(ticker, period="1mo", interval="1d"):
    try:
        data = yf.download(ticker, period=period, interval=interval)
        data["SMA"] = ta.trend.sma_indicator(data["Close"], window=14)
        data["EMA"] = ta.trend.ema_indicator(data["Close"], window=14)
        data["RSI"] = ta.momentum.rsi(data["Close"], window=14)
        macd = ta.trend.MACD(data["Close"])
        data["MACD"] = macd.macd()
        data["Signal"] = macd.macd_signal()
        data["Bollinger High"] = ta.volatility.bollinger_hband(data["Close"])
        data["Bollinger Low"] = ta.volatility.bollinger_lband(data["Close"])
        return data
    except Exception as e:
        st.error(f"Error fetching data: {e}")
        return None

data = fetch_data(ticker, period="1mo", interval=timeframe)

if data is not None:
    st.subheader(f"Price Data for {ticker}")
    st.dataframe(data.tail())

    # Plot Selected Indicator
    st.subheader(f"{indicator} Indicator Chart")
    if indicator == "SMA":
        st.line_chart(data[["Close", "SMA"]])
    elif indicator == "EMA":
        st.line_chart(data[["Close", "EMA"]])
    elif indicator == "RSI":
        st.line_chart(data[["RSI"]])
    elif indicator == "MACD":
        st.line_chart(data[["MACD", "Signal"]])
    elif indicator == "Bollinger Bands":
        st.line_chart(data[["Close", "Bollinger High", "Bollinger Low"]])

    # Signal Output
    st.subheader("Trading Signals")
    if indicator == "RSI":
        latest_rsi = data["RSI"].iloc[-1]
        if latest_rsi < 30:
            st.success("Buy Signal (Oversold)")
        elif latest_rsi > 70:
            st.error("Sell Signal (Overbought)")
        else:
            st.info("Neutral Signal")
    elif indicator == "MACD":
        latest_macd = data["MACD"].iloc[-1]
        latest_signal = data["Signal"].iloc[-1]
        if latest_macd > latest_signal:
            st.success("Buy Signal (MACD > Signal Line)")
        else:
            st.error("Sell Signal (MACD < Signal Line)")
    else:
        st.info("Signal generation only available for RSI and MACD.")
else:
    st.warning("No data available. Please check the ticker or timeframe.")
```