

Project Title:

Automatic Trading Bot

Contact Information:

Christopher Jean-Baptiste

Christopher.jean-baptiste45@bcmail.cuny.edu

Supervisor:

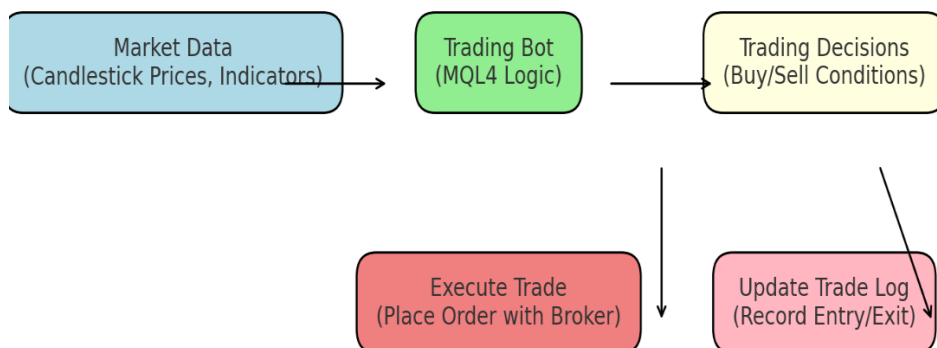
Sammyjoseph@me.com

Abstract:

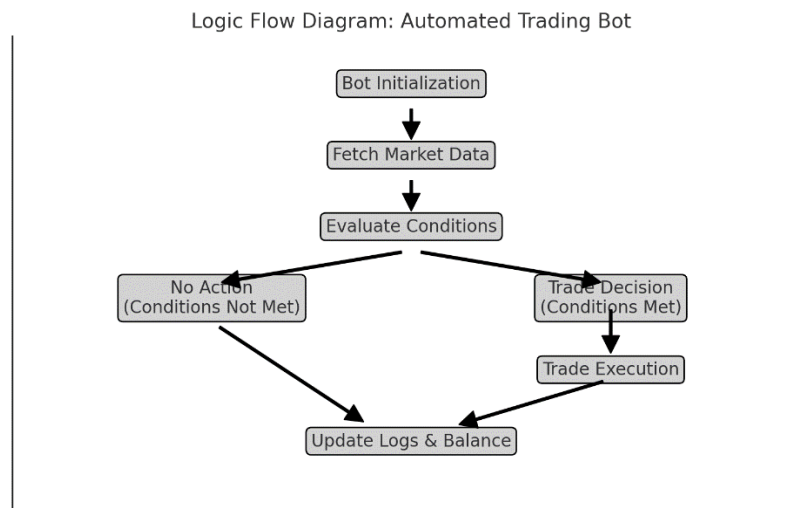
This project aims to design and implement an automatic trading bot for MetaTrader 4 (MT4), a popular platform for trading financial instruments such as currency pairs and futures. The bot will automate trading strategies based on predefined conditions and market analysis, ensuring timely and efficient execution of trades. The system will monitor live price feeds, analyze trends, and make buy/sell decisions according to customizable algorithmic rules. The project will help reduce manual effort and mitigate human error in trading decisions.

Tools:

- MetaTrader 4 platform (Build 1420)
- MQL4 Programming Language (Version 4)
- GitHub for version control
- Excel
- PC with MetaTrader 4 installed
- Access to MetaTrader 4 demo account for testing

Diagrams:

The trading bot continuously receives live market data, including candlestick prices and various technical indicators such as moving averages, RSI, etc. The trading bot is going to be programmed using the MQL4 language. It processes the incoming market data, and analyzes the data against predefined trading strategies or conditions. After analyzing the market data, the bot makes a trading decision. This decision could either be to buy, sell, or hold a position. If the bot decides to execute a trade, it places a buy or sell order and updates a trading log.



The bot initializes by setting up parameters and connects to MT4. It then fetches market data and evaluates conditions. From there, if conditions are met, the bot places buy or sell orders. If conditions are not met, the bot takes no action. Regardless the bot updates logs and repeats the cycle.

Tentative schedule:

Week	Task	Time Estimate (hours)
1-2	Research MQL4, gather technical requirements	10-12
3-4	Design trading strategies and risk management plan	10-12
5-6	Design core trading logic and indicator analysis	14-16
7-8	Implement UI for monitoring	14-16
9-10	Backtest with historical data	16-20
11-12	Implement improvements, demo live trading	16-20
13-14	Finalize code, documentation, and presentations	10

Data Sources:

- Real-time market data
- Historical data

Use Cases:

1. Stop-Loss and Take Profit

- Input: Active Buy / Sell order
- Action: Stop Loss is set 5 pips above or below entry price and take profit is set 30 pips above or below
- Expected Output: The bot will exit the trade if either stop-loss or take-profit conditions are met

2. Logging and trade history management

- Input: Trade details, account balance
- Action: When a trade is executed, the bot automatically logs all relevant details
- Expected Output: A complete and accurate log entry for every trade executed

3. 13 and 50 Moving Average Cross with Pullback Strategy

- Input: 13-period and 50-period EMA values calculated from the price data. Price movement (25-50 pips away from EMAs). Price pullback to the 50 EMA. Closing price relative to the 13 EMA after the pullback.
- Action: Bot executes the trade
- Expected Output: Bot places buy or sell order and logs the trade