PRACTICAL TASK

General Instructions:

- API documentation can be found over here: https://binance-docs.github.io/apidocs/futures/en/#general-info
- Binance TestNet Webscoket URL: wss://stream.binancefuture.com/ws
- Binance TestNet API URL: https://testnet.binancefuture.com
- [API Key: 0d1e94b104dd54fde98dec9a83f8916b1af3daa0c81c8c754b59ce3d62c8a00a]
- [API Secret: fd6302c060bdf02d8c5e369cc433eb802f6fa09b22317ae1a113f5ff86c40841]
- Binance TestNet URL: https://testnet.binancefuture.com/en/futures/BTCUSDT
- Sample code base can be found over here: https://github.com/Binance-docs/Binance_Futures_python

Primary Task Starts

Step 1: Create a Simple Django/ Flask App

- Create a Sample Django/ Flask App
- Create Websocket GET Route
- Create REST API POST Route

Step 2: GET Ticker via WebSocket API

- Connect to Binance Futures TestNet
- Fetch Bitcoin Price using WebSocket
- Display Price in Django/ Flask App
- Hint: Use Mini Ticker Stream API

Step 3: GET User Balance via REST API

- Connect to Binance Futures TestNet
- Fetch Account Details using REST API
- Display User Balance of USDT in Django/ Flask App
- Hint: Use Account Information v2 API

Step 4: POST Market Order via RESET API

- Connect to Binance Futures TestNet
- Fetch Bitcoin Price using WebSocket
- Post Market Order for $500 at a Leverage of 3x in BTCUSDT Pair
- Hint: Use New Order (Trade) API

Step 5: POST Limit Order via RESET API

- Connect to Binance Futures TestNet
- Post Limit Order in BTCUSDT Pair at 0.2% MarkUp of Purchase Price
- Hint: Use New Order (Trade) API

Primary Task Ends

Bonus (Optional) Task Starts

Bonus Step 1: RSI Order

- Place Market Order only if RSI is below 25

Bonus (Optional) Task Ends

===== xxx ===== xxx =====