

## 2. Crypto Trading Bot – Internship Assignment Report

### 1. Overview

This project implements a trading bot in Python using the Binance USDT-M Futures **Testnet**. It provides a CLI interface to execute **Market**, **Limit**, and **Stop-Market** orders. The bot uses the official `python-binance` library and follows a modular, clean structure with logging and input validation.

### 2. Features

- Market order execution (`market_orders.py`)
- Limit order execution (`limit_orders.py`)
- Stop-Market order (Advanced bonus) (`stop_limit.py`)
- CLI-based interaction
- Logging to `bot.log` with timestamps and error traces
- Input validation to handle incorrect values

### 3. Logs and Screenshots

The image shows a VS Code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'AKSHANSHI\_BINANCE\_BOT' with a 'src' directory containing 'advanced', 'bot.log', 'limit\_orders.py', and 'market\_orders.py'. The main editor window shows the code for 'stop\_limit.py' in the 'src/advanced' directory. The code defines a 'StopLimitOrderBot' class with an '\_\_init\_\_' method and a 'place\_stop\_limit\_order' method. The terminal shows the execution of 'stop\_limit.py', which prompts the user for API key, secret, trading pair, order side, quantity, stop price, and limit price. The script successfully places a stop-limit order for BTC/USD with a quantity of 0.01, a stop price of 57000, and a limit price of 56900.

```
src > advanced > stop_limit.py > ...
1 # src/advanced/stop_limit.py
2
3 from binance.client import Client
4 from binance.enums import *
5 import logging
6
7 # Setup logging
8 logging.basicConfig(
9     filename='bot.log',
10    level=logging.INFO,
11    format='%(asctime)s - %(levelname)s - %(message)s'
12 )
13
14 class StopLimitOrderBot:
15     def __init__(self, api_key, api_secret):
16         self.client = Client(api_key, api_secret)
17         self.client.FUTURES_URL = 'https://testnet.binancefuture.com/fapi'
18
19     def place_stop_limit_order(self, symbol, side, quantity, stop_price, limit_price):
20         try:
21             order = self.client.futures_create_order(
22                 symbol=symbol,
23                 side=side,
24                 quantity=quantity,
25                 stop_price=stop_price,
26                 limit_price=limit_price,
27                 order_type=ORDER_TYPE_STOP_MARKET
28             )
29         except Exception as e:
30             print(e)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
C:\Users\aksha\OneDrive\Desktop\Akshanshi_binance_bot\src\advanced>stop_limit.py
Traceback (most recent call last):
  File "C:\Users\aksha\OneDrive\Desktop\Akshanshi_binance_bot\src\advanced\stop_limit.py", line 6, in <module>
    from binance.enums import ORDER_TYPE_STOP_MARKET
ImportError: cannot import name 'ORDER_TYPE_STOP_MARKET' from 'binance.enums' (C:\Users\aksha\AppData\Local\Programs\Python\Python312\Lib\site-packages\binance\enums.py). Did you mean: "ORDER_TYPE_MARKET"?

C:\Users\aksha\OneDrive\Desktop\Akshanshi_binance_bot\src\advanced>stop_limit.py
[Script is running]
Enter your Binance Testnet API Key: sb4TkY0Me0yZ2wIntKIoFqbTfzQqUdZ4x0EoXrLBsg1XqCZANLZoN5j5yP30
Enter your Binance Testnet API Secret: D29xuKZ0vKu7D1dKPdufEMwGMS8t1zKIo0Et1MkXVRzdKgJYBFm1JJm9nynNABK0
Enter trading pair (e.g., BTCUSD): BTCUSD
Enter order side (BUY or SELL): BUY
Enter quantity to trade: 0.01
Enter stop price: 57000
Enter limit price: 56900
[Error placing order: APIError(code=-2015): Invalid API-key, IP, or permissions for action]
```

## 4. Challenges Faced

- Binance Testnet API now requires **\*\*KYC verification\*\*** to generate API keys.
- Encountered `APIError(code=-2015)` while using shared test credentials.
- Successfully tested structure, inputs, error handling, and logging despite the restriction.

## 5. Conclusion

All mandatory requirements were fulfilled:

- Market and Limit order bots built
- Bonus: Stop-Market order added
- Modular code, structured logs, and clear documentation provided