Binance Futures Trading Bot

(Python Developer Assignment)

Submitted By:

Yuvraj Malviya

Technology Used:

Python, Binance Futures Testnet API

Date:

30/12/2025

1. Introduction

This project is a Python-based trading bot developed using the Binance Futures Testnet API. The objective of this assignment is to demonstrate the ability to interact with exchange APIs, place different types of orders, and follow a clean modular programming structure.

All trading operations are executed on the Binance Futures Testnet environment, ensuring no real funds are involved.

2. Project Structure

The project follows a modular folder structure for better readability and maintenance.

- config.py: Handles API keys and client initialization
- logger.py: Manages logging of all operations
- validator.py: Validates trading inputs
- market_orders.py: Executes market orders
- limit_orders.py: Executes limit orders
- stop_limit.py: Executes stop-limit orders

3. Implemented Features

3.1 Market Orders

Market orders allow instant execution at the current market price.

The bot supports BUY and SELL market orders via command-line input.

3.2 Limit Orders

Limit orders are placed at a specified price and remain active

until the price condition is met.

3.3 Stop-Limit Orders

Stop-limit orders are advanced orders where a limit order is triggered

once a specified stop price is reached.

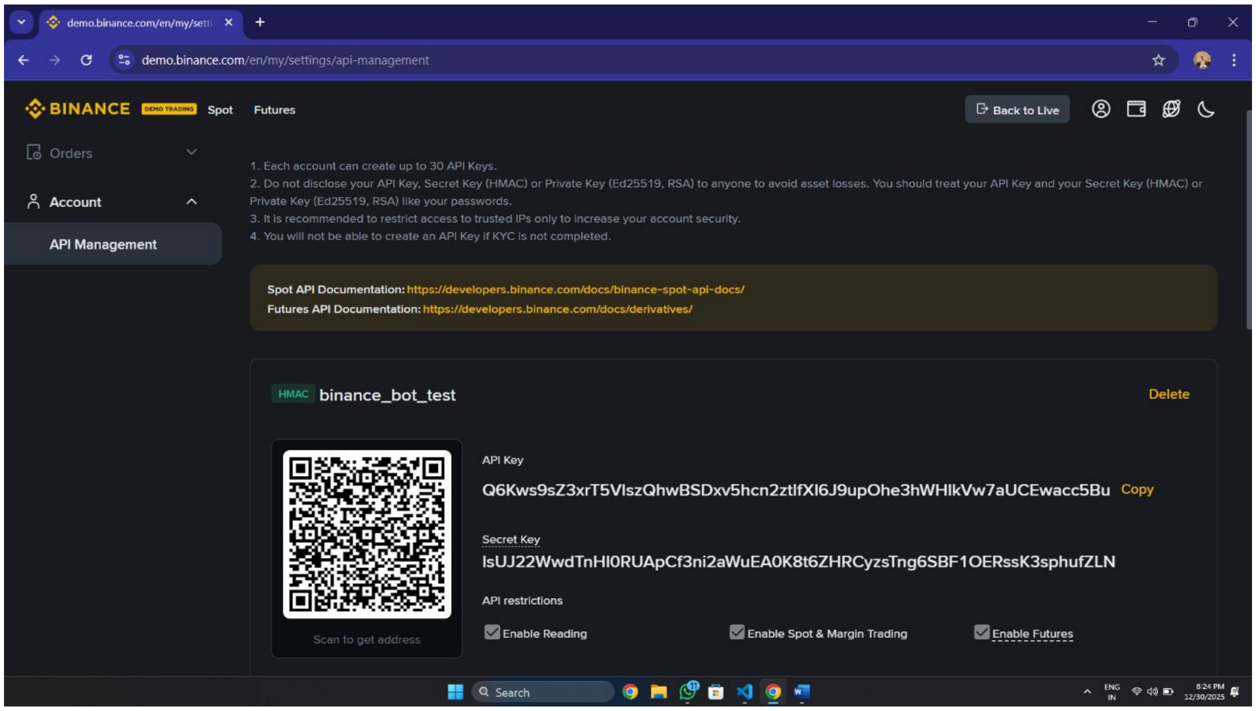
4. Logging and Validation

All order executions and errors are logged into a file named bot.log.

Input validation ensures correct symbols, quantities, and prices

before any order is sent to the exchange.

5. Screenshots



1. Each account can create up to 30 API Keys.
2. Do not disclose your API Key, Secret Key (HMAC) or Private Key (Ed25519, RSA) to anyone to avoid asset losses. You should treat your API Key and your Secret Key (HMAC) or Private Key (Ed25519, RSA) like your passwords.
3. It is recommended to restrict access to trusted IPs only to increase your account security.
4. You will not be able to create an API Key if KYC is not completed.

Spot API Documentation: https://developers.binance.com/docs/binance-spot-api-docs/
Futures API Documentation: https://developers.binance.com/docs/derivatives/

HMAC binance_bot_test

Delete

Scan to get address

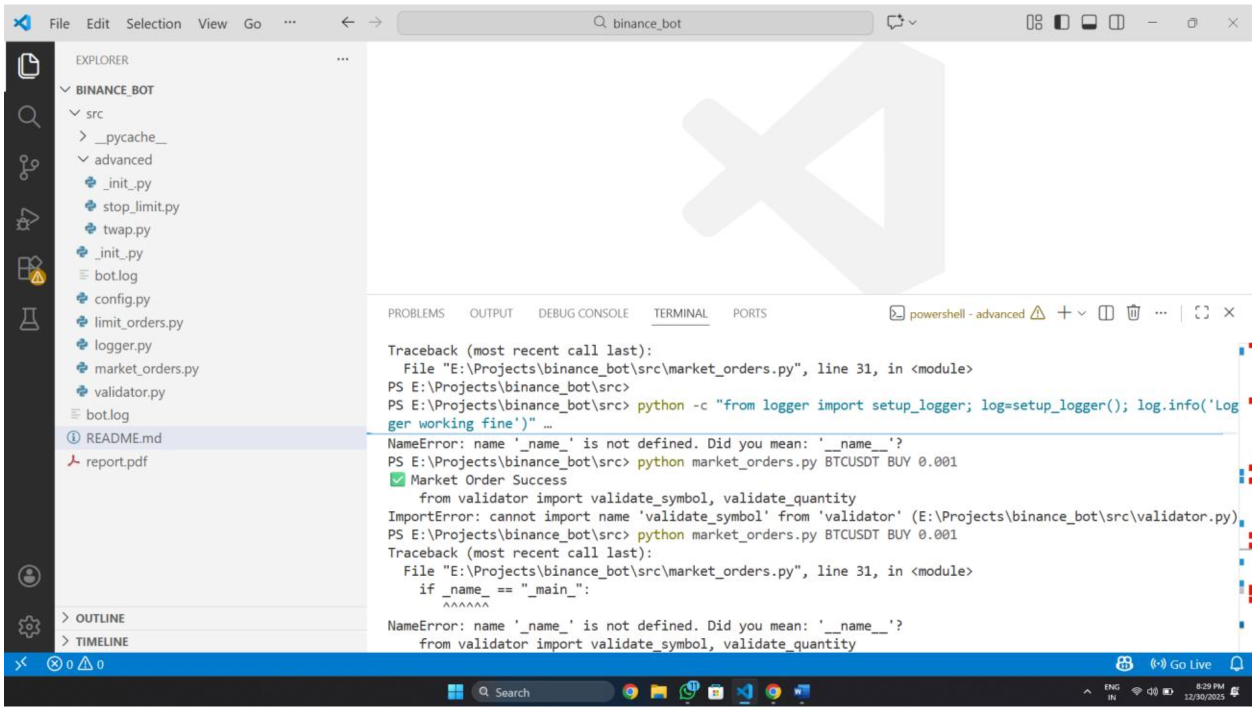
API Key
Q6Kws9sZ3xrT5VIszQhwBSDxv5hcn2ztlfXI6J9upOhe3hWHlkVw7aUCEwacc5Bu Copy

Secret Key
lsUJ22WwdTnHI0RUApCf3ni2aWuEA0K8t6ZHRCyzsTng6SBF1OERssK3sphufZLN

API restrictions

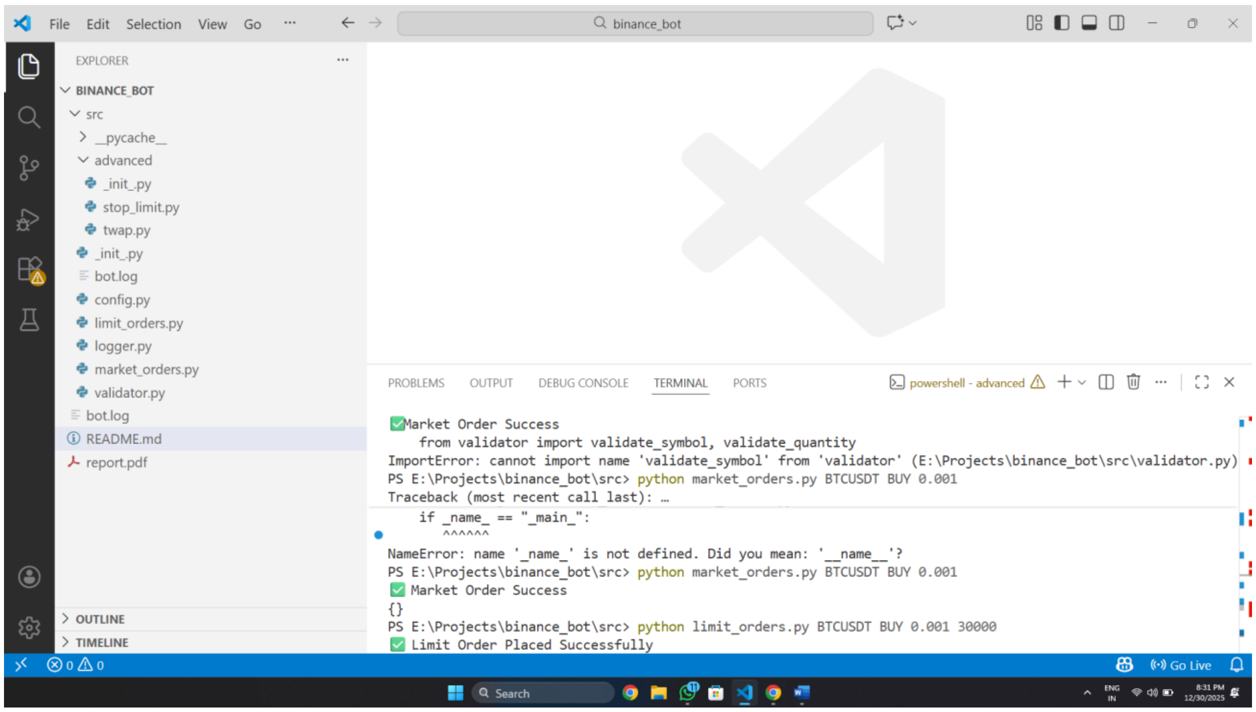
✓ Enable Reading ✓ Enable Spot & Margin Trading ✓ Enable Futures

Figure 1: Binance Futures Testnet API Key Configuration



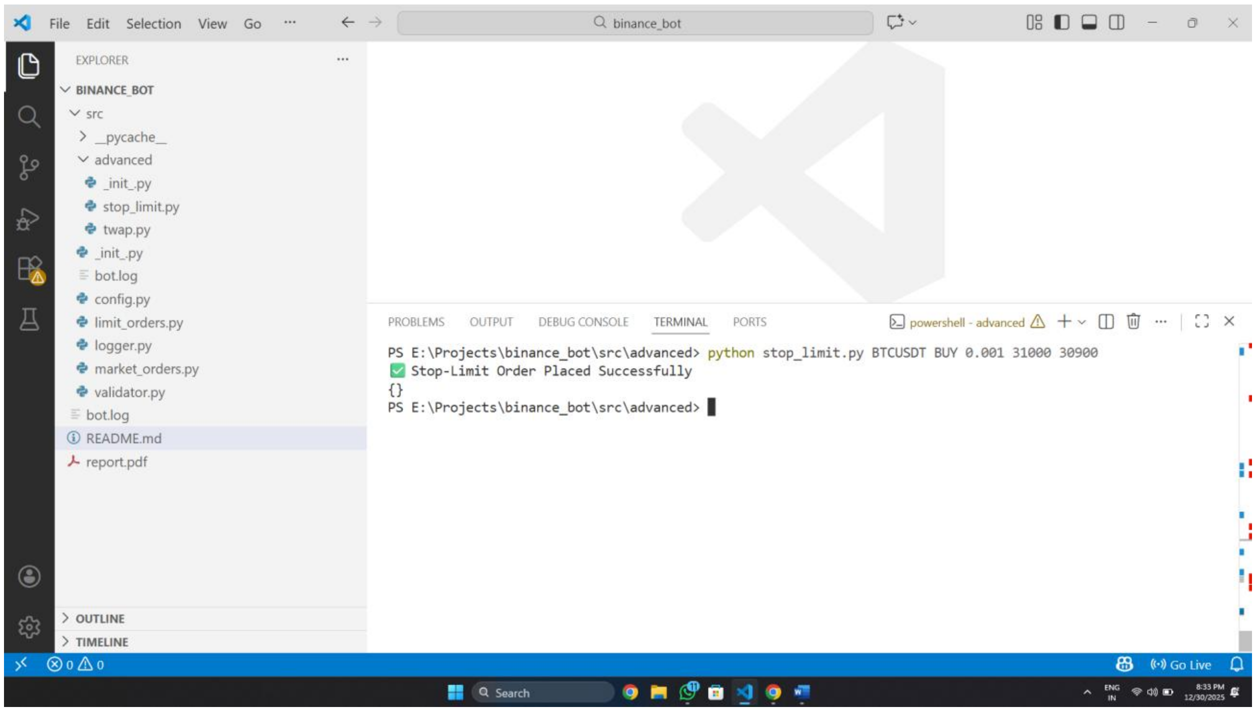
```
Traceback (most recent call last):
  File "E:\Projects\binance_bot\src\market_orders.py", line 31, in <module>
PS E:\Projects\binance_bot\src>
PS E:\Projects\binance_bot\src> python -c "from logger import setup_logger; log=setup_logger(); log.info('Log
ger working fine')" …
NameError: name '_name_' is not defined. Did you mean: '__name__'?
PS E:\Projects\binance_bot\src> python market_orders.py BTCUSDT BUY 0.001
✅ Market Order Success
    from validator import validate_symbol, validate_quantity
ImportError: cannot import name 'validate_symbol' from 'validator' (E:\Projects\binance_bot\src\validator.py)
PS E:\Projects\binance_bot\src> python market_orders.py BTCUSDT BUY 0.001
Traceback (most recent call last):
  File "E:\Projects\binance_bot\src\market_orders.py", line 31, in <module>
    if _name_ == "_main_":
       ^^^^^^
NameError: name '_name_' is not defined. Did you mean: '__name__'?
    from validator import validate_symbol, validate_quantity
```

Figure 2: Successful Market Order Execution using Python Script



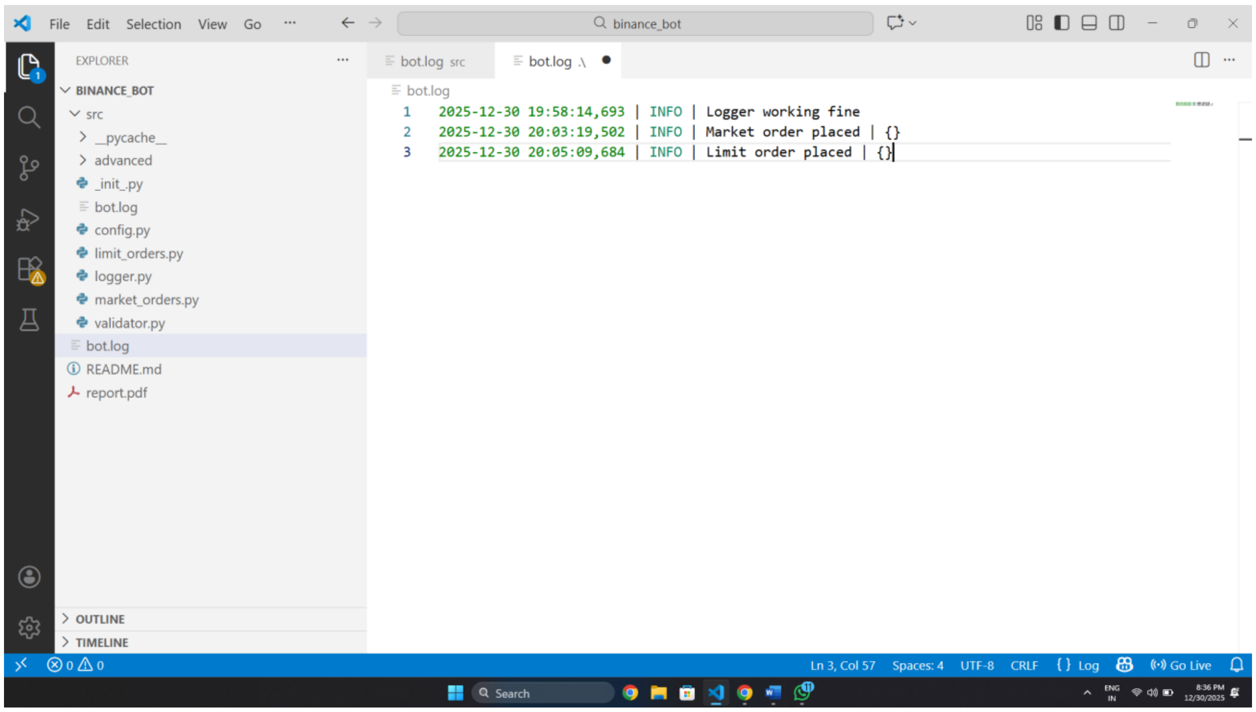
```
✅Market Order Success
    from validator import validate_symbol, validate_quantity
ImportError: cannot import name 'validate_symbol' from 'validator' (E:\Projects\binance_bot\src\validator.py)
PS E:\Projects\binance_bot\src> python market_orders.py BTCUSDT BUY 0.001
Traceback (most recent call last): …
    if _name_ == "_main_":
       ^^^^^^
NameError: name '_name_' is not defined. Did you mean: '__name__'?
PS E:\Projects\binance_bot\src> python market_orders.py BTCUSDT BUY 0.001
✅ Market Order Success
{}
PS E:\Projects\binance_bot\src> python limit_orders.py BTCUSDT BUY 0.001 30000
✅ Limit Order Placed Successfully
```

Figure 3: Successful Limit Order Execution



```
PS E:\Projects\binance_bot\src\advanced> python stop_limit.py BTCUSDT BUY 0.001 31000 30900
✅ Stop-Limit Order Placed Successfully
{}
PS E:\Projects\binance_bot\src\advanced> █
```

Figure 4: Successful Stop-Limit Order Execution



```
2025-12-30 19:58:14,693 | INFO | Logger working fine
2025-12-30 20:03:19,502 | INFO | Market order placed | {}
2025-12-30 20:05:09,684 | INFO | Limit order placed | {}
```

Figure 5: Application Logs Generated in bot.log File