Simplified Binance Trading Bot — Report

Author: Mohit Sharma
Date: October 2025

1. Overview

The Simplified Binance Trading Bot is a Python-based project built for the Binance Futures Testnet. It interacts with Binance's REST API using the python-binance library. The bot places Market and Limit orders, supports Buy and Sell actions, logs all activities, and can handle errors gracefully. Additionally, optional advanced order types like OCO and TWAP are supported.

2. Architecture

User Input (CLI) → BasicBot (Core class) → Binance Futures Testnet API → Response + Logging

Modules:
• bot_core.py – Core API logic
• market_orders.py – Market order handling
• limit_orders.py – Limit order handling
• advanced/oco.py – OCO order example
• advanced/twap.py – TWAP strategy example

Supporting Files:
• bot.log – Logs of execution
• README.md – Usage instructions
• report.pdf – Project summary

3. Features

✔■ Place Market & Limit orders on Binance Futures Testnet
✔■ Support both BUY and SELL sides
✔■ Clean, reusable object-oriented design
✔■ Command-line based user input
✔■ Detailed logging and error handling

✔■ Optional advanced orders (OCO, TWAP)

4. Example Workflow

1■■ Run:
python src/market_orders.py

2■■ Input details:
Enter API Key: ********** Enter Secret Key: ********** Enter symbol: BTCUSDT Enter side (BUY/SELL): BUY Enter order type (MARKET/LIMIT): MARKET Enter quantity: 0.001
3■■ Output:
■ Order placed successfully! Order ID: 123456789 Symbol: BTCUSDT Side: BUY Status: FILLED Price: 67000.00

5. Sample Log Entry

[2025-10-24 13:42:02] INFO - MARKET ORDER: BUY 0.001 BTCUSDT placed successfully. [2025-10-24 13:42:03] INFO - Order ID: 123456789 | Status: FILLED | Price: 67000.00

6. Screenshots

• CLI showing order placement
• Binance Testnet dashboard showing executed order
• Log file with success entries
(Add screenshots before submission)

7. Key Learnings

• Learned how to authenticate and use Binance REST APIs
• Understood signed requests and payload validation
• Practiced modular and object-oriented code structure
• Implemented real-time logging and error tracking

8. Conclusion

The project fulfills all requirements: market & limit orders, buy/sell functionality, structured logging, and modular code. Optional OCO and TWAP demonstrate scalability for advanced trading strategies. It provides a strong foundation for safe experimentation and future automation on live exchanges.