Assignment Report
Binance Trading Bot

Prepared by: Pratik Raut

Date: 01-10-25

1. Introduction

This project is about implementing a simple Binance Trading Bot using Python. The bot supports multiple order types and a TWAP strategy, using the Binance Futures API. It is designed with a clean folder structure, logging, and command-line usage.

2. Project Structure

The project follows the required folder structure:

project/

├── src/

│ ├── market_orders.py

│ ├── limit_orders.py

│ └── advanced/

│ ├── oco.py

│ └── twap.py

├── bot.log

├── README.md

└── report.pdf

- market_orders.py → Handles market buy/sell orders. - limit_orders.py → Handles limit orders at a

specific price. - advanced/oco.py → Implements OCO (One Cancels the Other) orders. advanced/twap.py → Implements TWAP strategy for splitting orders. - bot.log → Log file storing every action and error. - README.md → Instructions for usage. - report.pdf → Documentation of the project.

3. Code Explanation

3.1 Market Orders

Immediate execution at current market price. Command: python src/market_orders.py BTCUSDT BUY 0.001

3.2 Limit Orders

Executes only at a specific target price. Command: python src/limit_orders.py BTCUSDT SELL 0.001 60000

3.3 OCO Orders

Places a take profit and stop loss together. If one executes, the other is canceled. Command: python src/advanced/oco.py BTCUSDT SELL 0.001 62000 59000 58500

3.4 TWAP Strategy

Splits a large order into smaller chunks executed over time. Command: python src/advanced/twap.py BTCUSDT BUY 0.05 5 10 → Places 5 buy orders of 0.01 BTC, 10 seconds apart.

4. Logging

All actions are stored in bot.log. Example log entry: 2025-10-01 22:15:41 [INFO] TWAP order 5/5 placed: {...}

5. Testing

1. Connection Test python src/test_connection.py 2. Market Order Test python src/market_orders.py BTCUSDT BUY 0.001 3. TWAP Order Test python src/advanced/twap.py BTCUSDT BUY 0.05 5 10 4. Checking Logs cat bot.log

6. Screenshots (to include)

- CLI execution of market_orders.py - CLI execution of limit_orders.py - CLI execution of oco.py CLI execution of twap.py - Sample bot.log output

7. Conclusion

This assignment demonstrates how to use the Binance Futures API in Python for automated trading. The bot supports essential trading operations (Market, Limit, OCO, TWAP) with logging and modular structure. It provides a foundation for more advanced strategies such as grid trading, scalping, or arbitrage.