Report — Binance Futures Order Bot

Developer Details

Name: Sumedha Hemadri

Role: Python Developer Intern Applicant

Email: sumedhahemad@gmail.com

Date: 5-10-2025

Objective

To develop a CLI-based trading bot for Binance USDT-M Futures that supports both core and advanced order types, ensuring robust validation, structured logging, and clear documentation.

Key Responsibilities Covered

Responsibility	Implemented
Market Orders	Yes
Limit Orders	Yes
Stop-Limit Orders	Yes (Simulated)
OCO Orders	Yes (Simulated)
TWAP	Yes
Grid Orders	Yes
Validation & Logging	Yes
Report & Documentation	Yes

Technologies Used

Technology	Purpose
Python 3.12	Programming language
Logging	Structured logging to bot.log
JSON	Structured log entries
Random, Time	Order simulation & timestamps
VS Code / Jupyter	Development environment

Folder Structure

```
sumedha_binance_bot_project/
│
├── src/
│   ├── common_utils.py
│   ├── run_demo.py
```

```
├── bot.log
├── README.
└── Report.pdf
```

Features Implemented

Core Orders: Market and Limit orders with validation.

Advanced Orders: Stop-Limit, OCO, TWAP, and Grid orders simulated.

Validation & Logging: Structured JSON logs in bot.log.

Sample Execution Output

=== Market Order === ('symbol': 'BTCUSDT', 'type': 'MARKET', 'side': 'BUY', 'orderId': 458112, 'status': 'NEW', 'quantity': 0.001} === Limit Order === {'symbol': 'BTCUSDT', 'type': 'LIMIT', 'side". 'SELL', 'orderId". 489231, 'status': 'NEW', 'quantity". 0.001, 'price': 60000}

Sample Log Entries (bot.log)

("timestamp": "2025-10-05 14:12:30", "level": "INFO", "action": "MARKET ORDER SUCCESS", "payload": {"symbol": "BTCUSDT", "side": "BUY"}} {"timestamp": "2025-10-05 14:12:33", "level": "INFO", "action": "LIMIT ORDER SUCCESS", "payload": ("symbol": "BTCUSDT", "side": "SELL", "price": 60000}}

How to Run the Bot

Terminal: python src/run_demo.py

Jupyter: Run cells sequentially to view simulated output.

Logs: Check bot.log after execution.

Evaluation Alignment

Criteria	Weight	Status
Basic Orders	50%	Done
Advanced Orders	30%	Done
Logging & Errors	10%	Done
Documentation	10%	Done

Conclusion

The Binance Futures Order Bot fulfills all the assignment criteria by simulating multiple order types, incorporating robust validation, and maintaining detailed structured logs. It ensures modularity, reproducibility, and readiness for real Binance API integration.

References

- Binance Futures API Docs: https://binance-docs.github.io/apidocs/futures/en/

- Python Standard Library Documentation (logging, json, os, random)
- Assignment Document Provided by Company

-