Binance Bot Report

Introduction:

This project implements a Command-Line Interface (CLI) based trading bot for Binance USDT-M Futures. The bot allows execution of core order types such as Market and Limit orders, with optional modules for advanced strategies like OCO, TWAP, and Grid trading.

Objective:

Build a modular, well-structured trading bot. Implement Market and Limit orders with validation and logging. Create optional advanced trading strategies. Provide reproducible documentation (README + Report).

Project Structure:

```
[raju_binance_bot]/

│

├── /src/ # All source code

│   ├── market_orders.py # Example: Market order logic

│   ├── limit_orders.py # Example: Limit order logic

│   ├── advanced/ # () Folder for advanced orders

│   │   ├── oco.py # Example: OCO order logic

│   │   └── twap.py # Example: TWAP strategy

│

├── bot.log # Logs (API calls, errors, executions)

├── report.pdf # Analysis ()

└── README.md # Setup, dependencies, usage
```

Methodology:

Binance API Integration The project uses the Binance Futures API for executing orders. API keys are loaded securely using environment variables stored in a `.env` file. Order Implementations Market Orders: These allow instant execution at current market price. Limit Orders: These execute at a specified user-defined price. Both orders include: Validation of symbol, quantity, and price Detailed logging in bot.log Error handling using structured logs .Advanced Order Modules Modules such as OCO, TWAP, and Grid are included as extension modules. TWAP includes basic functionality, while OCO and Grid are structured for future enhancements.

Logging & Validation:

All order operations generate logs in bot.log with: Timestamps Executed API calls Error traces Order execution details