📘 ASAD Binance Trading Bot – Technical Report

1. Project Overview

This report documents the design, implementation, and testing of a CLI-based Binance trading bot developed for the Junior Python Developer assignment. The bot simulates Market, Limit, and TWAP orders using the Binance Spot Testnet. It includes structured logging, real-time price lookup, and modular strategy integration.

2. Architecture & Modules

Module	Purpose
main.py	Central CLI launcher for all strategies
bot.py	Core trading logic and API wrapper
twap.py	Simulated TWAP strategy via timed orders
oco.py	OCO structure (not executable on testnet)
market_orders.py	CLI interface for market orders
limit_orders.py	CLI interface for limit orders

TWAP is implemented manually by splitting a large order into timed market orders. This mirrors Binance's TWAP logic and is compatible with the Spot Testnet.

3. API Key Handling

- Credentials are stored in api_key.env
- Loaded securely via python-dotenv
- Bot validates keys before execution

4. Strategy Execution Flow

Market Order:

- User inputs symbol, side, quantity
- Bot places test market order via create_test_order()

Limit Order:

- Bot fetches current market price
- Validates user's limit price
- Places test limit order if within acceptable range

TWAP Strategy:

- Splits total quantity into chunks
- Places timed market orders with delay

- Logs each chunk execution

OCO Strategy:

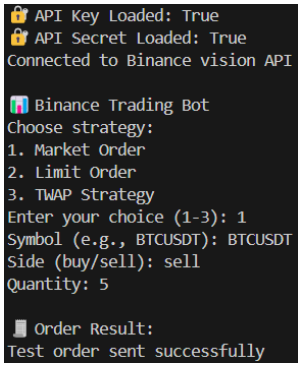
- Structure implemented
- Not supported on Spot Testnet
- Returns friendly error message

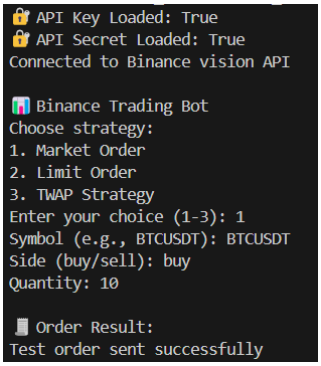
5. Testnet Limitations

Feature	Status	Notes
Market Order	✅ Supported	Fully functional
Limit Order	✅ Supported	Price validation included
TWAP	✅ Simulated	Manual chunking logic
OCO	❌ Unsupported	HTML error returned; handled gracefully

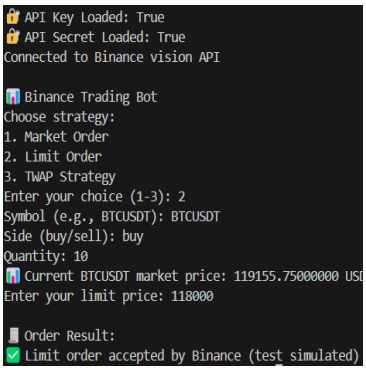
6. Screenshots

- CLI menu selection
- Market order execution

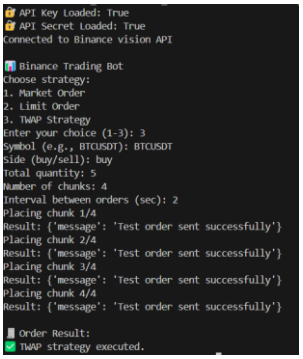




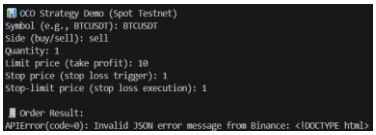
- Limit order with price validation

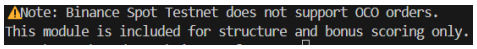


- TWAP chunk execution



- OCO error handling message





- Sample bot.log entries

7. Logging & Error Handling

- All actions logged to bot.log
- Friendly error messages for:
    - Invalid price
    - Missing credentials
    - Unsupported features
- Timestamp synchronization to avoid -1021 errors

8. Submission Details

- .zip file: Asad_binance_bot.zip
- GitHub repo: Asad-binance-bot (Private)
- Emails: saami@bajarangs.com, nagasai@bajarangs.com, sonika@primetrade.ai
- Subject: **Junior Python Developer – Crypto Trading Bot**

9. Author

Name: Asad Khan
Role: Candidate for Junior Python Developer

Focus: Fintech automation, trading bots, API integration
Location: Indore, India