Binance Futures Testnet Trading Bot - Final Report (with Mock Screenshots)

Purpose & Scope

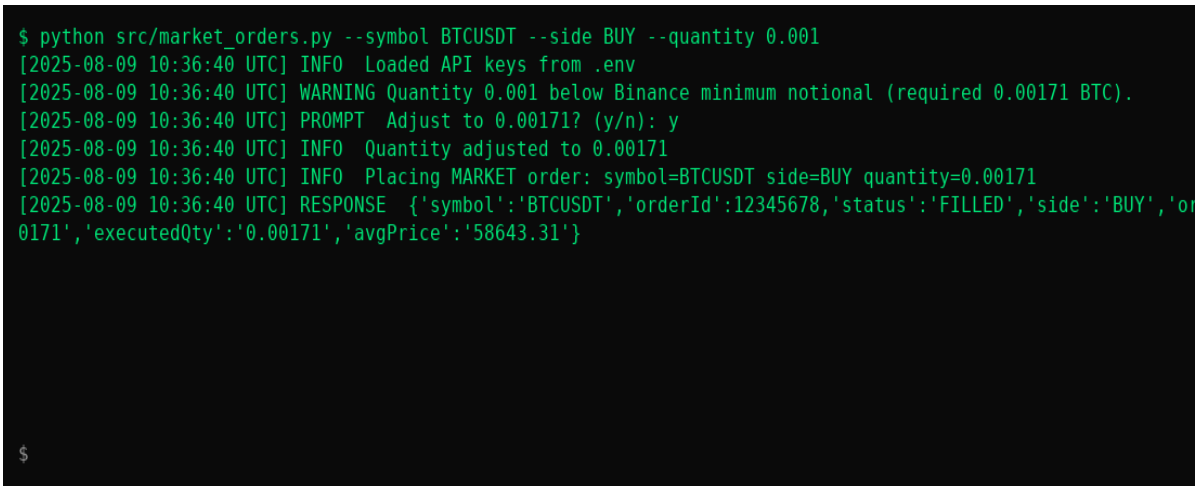
Analytical report describing the bot's design, responsibilities, and limitations.

System Architecture

Modular CLI scripts for each order type (market, limit, advanced). Shared utilities handle .env loading, authentication, validation, and logging.

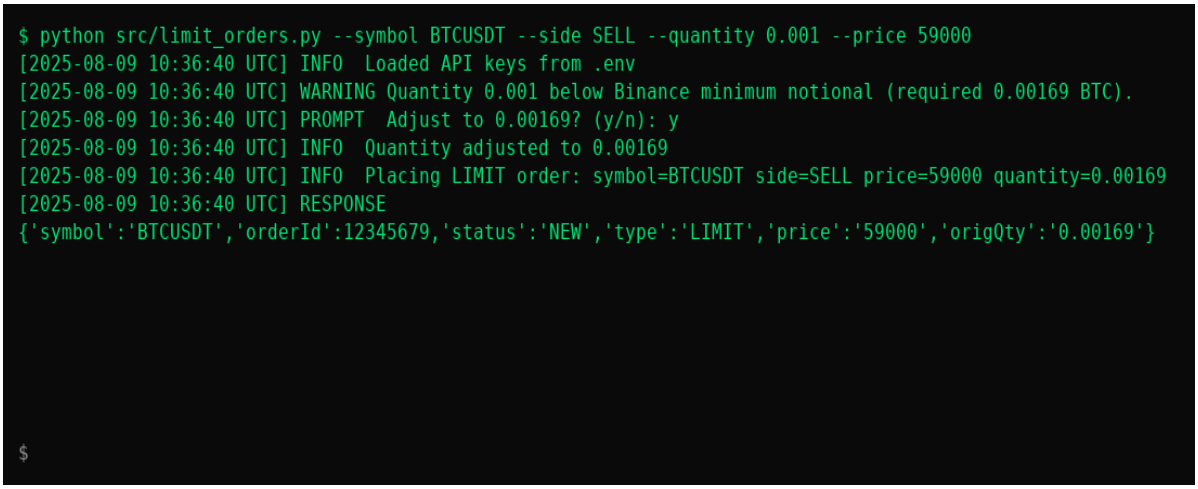
Market Orders

Market orders execute immediately at market price. Below is a mock terminal run showing quantity adjustment (+confirmation) and a filled market order.



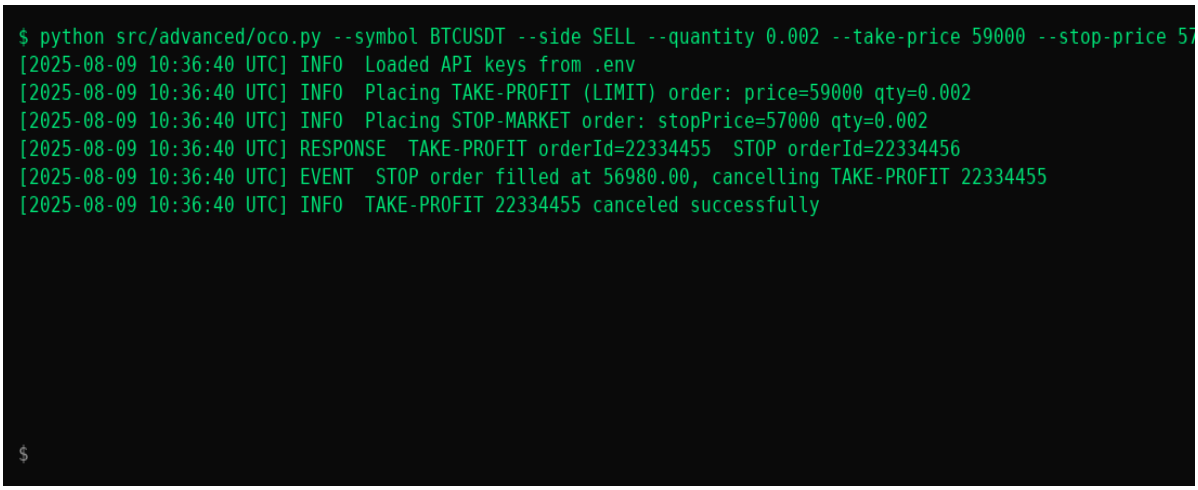
Limit Orders

Limit orders execute at a specified price. The example demonstrates the minimum notional adjustment prompt and a NEW limit order response.



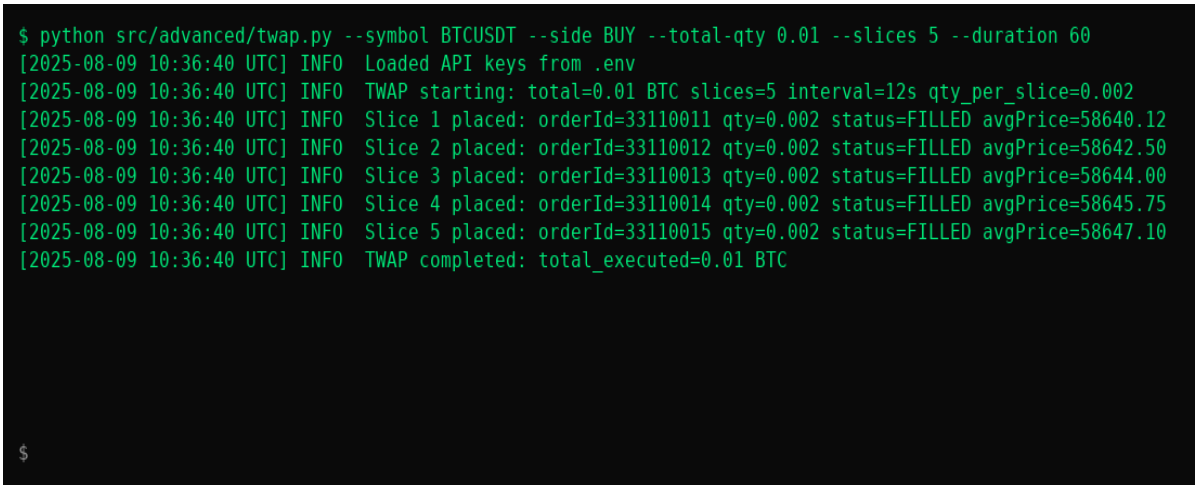
OCO Orders

OCO places a take-profit limit and a stop-market stop-loss; when one triggers the other is cancelled. The mock shows a stop triggering and cancelling the TP.



TWAP Orders

TWAP breaks a large order into slices; the mock shows 5 slices executed sequentially with individual fill reports.



Validation & Error Handling

The bot validates symbol, quantity (min notional), and logical relationships (e.g., OCO stop vs take), and logs API errors.

Logging

All requests, responses, adjustments, and errors are logged to bot.log for auditability.

Limitations & Recommendations

Testnet-only by default; recommend websocket integration, exchange filter fetching, and more advanced strategy tooling for production.

Conclusion

A robust foundation for futures automation in a test environment, suitable for extension and production hardening.