Application Task – Python Developer (Trading Bot on Binance Futures Testnet) : Estimated time less than 60 minutes

Thanks for applying. As the first step in our hiring process, please complete the task below. Candidates who meet the acceptance criteria will be shortlisted for an interview.

Application Task: Build a Simplified Trading Bot (Binance Futures Testnet)

Objective

Create a small Python application that can place orders on **Binance Futures Testnet (USDT-M)** and provide a clean, reusable structure with proper logging and error handling.

Setup

1. Register and activate a **Binance Futures Testnet** account.
2. Generate API credentials.
3. Use this testnet base URL for all API interactions:
   `https://testnet.binancefuture.com`
4. You may use either:
   - `python-binance` library, **or**
   - direct REST calls (requests/httpx)

Core Requirements (must-have)

Language: Python 3.x

Your app must:

1. Place **Market** and **Limit** orders on **Binance Futures Testnet (USDT-M)**
2. Support both sides: **BUY** and **SELL**
3. Accept and validate user input via **CLI** (e.g., argparse / Typer / Click):
   - symbol (e.g., BTCUSDT)
   - side (BUY/SELL)
   - order type (MARKET/LIMIT)
   - quantity
   - price (required for LIMIT)
4. Print clear output:
   - order request summary
   - order response details (orderId, status, executedQty, avgPrice if available)
   - success/failure message
5. Implement:
   - structured code (separate client/API layer and command/CLI layer)

- logging of API requests, responses, and errors to a log file
- exception handling (invalid input, API errors, network failures)

Deliverables

Please submit:

1. A public GitHub repository (preferred) OR a zip folder containing:
   - source code
   - `README.md` with:
     - setup steps
     - how to run examples
     - any assumptions
   - `requirements.txt` (or `pyproject.toml`)
2. Log files from at least:
   - one MARKET order
   - one LIMIT order

Bonus (optional — choose any one)

- Add a third order type: **Stop-Limit / OCO / TWAP / Grid**
- Add an enhanced CLI UX (menus, prompts, validation messages)
- Add a lightweight UI (optional)

Suggested Project Structure (optional)

```
trading_bot/
  bot/
    __init__.py
    client.py        # Binance client wrapper
    orders.py        # order placement logic
    validators.py    # input validation
    logging_config.py
  cli.py             # CLI entry point
  README.md
  requirements.txt
```

Evaluation Criteria (what we grade)

- Correctness: places orders successfully on testnet
- Code quality: readability, structure, reuse

- Validation + error handling
- Logging quality (useful, not noisy)
- Clear README + runnable instructions

How to Apply

Email your **resume + submission link** (GitHub/zip) + **log files** to:

- joydip@anything.ai
- chetan@anything.ai
- hello@anything.ai
- CC: sonika@anything.ai