Assignment: Binance Futures Order Bot

Objective

Develop a CLI-based trading bot for Binance USDT-M Futures that supports multiple order types with robust logging, validation, and documentation.

Key Responsibilities

1. Core Orders (Mandatory)

- Market Orders
- Limit Orders

2. Advanced Orders (Bonus – Higher Priority in Evaluation)

- Stop-Limit Orders (e.g., trigger a limit order when a stop price is hit)
- OCO (One-Cancels-the-Other) (e.g., place a take-profit and stop-loss simultaneously)
- TWAP (Time-Weighted Average Price) (e.g., split large orders into smaller chunks over time)
- Grid Orders (e.g., automated buy-low/sell-high within a price range)

3. Validation & Logging

- Validate inputs (symbol, quantity, price thresholds).
- Log all actions (order placement, errors, executions) in a structured log file.

Submission Guidelines

1. File Structure

Submit a single `.zip` file named `[your_name]_binance_bot.zip` with this structure:

```
[project_root]/
|
├── /src/              # All source code
|   ├── market_orders.py  # Example: Market order logic
|   ├── limit_orders.py  # Example: Limit order logic
|   ├── advanced/        # (Bonus) 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 (screenshots, explanations)
└── README.md          # Setup, dependencies, usage
```

2. GitHub Submission

- Push your code to a private GitHub repo.
- Follow the same structure as the `.zip` file.
- Add collaborators: `[your_instructor_github_username]`.
- Repo Naming: `[your_name]-binance-bot` (e.g., `alice-binance-bot`).

Evaluation Criteria

Criteria	Weight	Notes
Basic Orders	50%	Market/limit orders with validation.
Advanced Orders	30%	Stop-Limit, OCO, TWAP, etc. Higher priority if implemented.
Logging & Errors	10%	Structured `bot.log` with timestamps and error traces.
Report & Docs	10%	Clear `README.md` and `report.pdf` with screenshots.

Note: Submissions with advanced orders and proper logs will rank higher.

Resources

- Binance Futures API Docs: https://binance-docs.github.io/apidocs/futures/en/
- Historical Data: https://drive.google.com/file/d/1IAfLZwu6rJzyWKgBToqwSmmVYU6VbjVs/view (Optional for testing)
- Fear & Greed Index: https://drive.google.com/file/d/1PgQC0tO8XN-wqkNyghWc_-mnrYv_nhSf/view (Bonus integration)

Final Notes

1. No hardcoded filenames: Use descriptive names (e.g., `grid_strategy.py` instead of `task3.py`).
2. Reproducibility: Ensure your `README.md` includes:
   - API setup instructions.
   - How to run the bot (e.g., `python src/market_orders.py BTCUSDT BUY 0.01`).
3. Deadline: Submit the `.zip` and GitHub repo link by [date].

Questions? Reach out to [your contact email].

Key Changes from Original

- No code snippets: Only conceptual examples (e.g., "OCO orders").
- Flexible naming: Emphasized descriptive filenames over hardcoded ones.
- GitHub integration: Added private repo instructions.
- Prioritization clarity: Highlighted advanced orders as a key differentiator.