

Project Report

1. What This Project Is About

This project was about building two main things:

- **A simple computer program (bot) for trading:** This bot helps you buy and sell digital money (like Bitcoin) on a special practice platform called Binance Testnet.
- **A cool website that explains the bot:** This website shows you how the bot works in a fun, easy-to-understand way, using pictures and interactive parts. Our main goal was to show how to make a trading bot using Python, and then how to explain complicated tech stuff clearly on a website.

The Two Main Parts:

- **Trading Bot (Python Program):** This is a program you run on your computer. It can place different types of orders (like "buy now" or "buy at a certain price") on a fake trading platform. It also keeps a record of everything it does.
- **Explainer Website (Interactive Page):** This is a single web page that uses charts and diagrams to make it super easy to understand what the trading bot does and how to use it.

2. How We Built It :

The Trading Bot (Python Program)

What We Used:

- **Python:** The programming language.
- `python-binance`: A special tool (library) that helps Python talk to Binance's trading platform.

How It Works:

- **Practice Trading (Testnet):** The bot only works on Binance's "Testnet." This is like a practice field where you can trade with fake money, so you don't lose real money while learning.
- **Placing Orders:**
 - ✓ **Market Orders:** Buy or sell immediately at the current best price.
 - ✓ **Limit Orders:** Set a specific price you want to buy or sell at. The order waits until that price is hit.
 - ✓ **Stop-Market Orders:** This order waits until a certain "stop price" is reached, then it turns into a "Market Order" and buys/sells immediately.

- **Buying and Selling:** The bot can do both.
- **Checking Your Money:** It can tell you how much fake USDT (a type of digital money) you have in your practice account.
- **Understanding Trading Rules:** The bot can look up Binance's rules for each type of digital money (like how many decimal places prices can have, or the smallest amount you can trade). This helps it avoid mistakes.
- **Talking to the Bot (CLI):** You type commands into your computer's command window to tell the bot what to do, like entering your secret keys or placing an order.
- **Keeping Records (Logging):** The bot writes down everything it does (every command, every response from Binance, and any errors) into a file called `trading_bot.log`. This is super helpful for checking what happened.
- **Handling Problems (Error Handling):** If something goes wrong (like you type a wrong command, or Binance sends an error message), the bot tries to understand the problem and tells you about it, instead of just crashing.

3. Problems We Solved :

Building this project had a few bumps:

- **"Wrong Address" Error for Binance:**
 - **Problem:** At first, the bot couldn't connect to the Binance Testnet because we were telling it the wrong web address.
 - **Solution:** We found out that the `python-binance` tool automatically knows the Testnet address if you just tell it `testnet=True`, so we removed the manual address.
- **Confusing Instructions for Users:**
 - **Problem:** People using the bot in the command window sometimes typed the wrong things, like typing "BUY" when the bot expected "MARKET" for the order type.
 - **Solution:** We made the instructions clearer in the command window. More importantly, we added smart checks inside the bot. Now, before the bot even tries to send an order to Binance, it checks if your quantity and price follow Binance's rules. This stops a lot of errors before they happen.

4. What We Achieved

We successfully created two main things:

1. **A Working Trading Bot (Python):** A program that can perform basic trading actions on the Binance Testnet, complete with good record-keeping.
2. **An Explainer Website (HTML/CSS/JS):** A single, interactive web page that makes the bot's features and how-to guide very easy to understand and visually appealing.

5. What's Next (Future Ideas) :

This project is a good start. Here are some ideas for making it even better:

- **More Order Types:** Add even more complex ways to trade, like "One-Cancels-the-Other" orders.
- **Manage Your Trades:** Add features to see all your open trades and change how much risk you're taking.
- **Live Data:** Make the bot get real-time price updates so it can react faster.
- **Easy-to-Use App:** Build a proper app with buttons and menus instead of just typing commands.

6. Conclusion :

This project showed that we can build useful Python tools for finance and then explain them clearly using modern web design. By fixing problems along the way, we learned a lot about making software that works well and is easy for people to understand. The trading bot and the explainer website together show a complete solution, combining clever programming with great presentation.