Name – Sagar Gurung

Position applied for – Python Developer

Project Documentation: Sagars Binance Futures Order Bot

Introduction

Sagars Binance Futures Order Bot is a command-line tool designed to automate order placement on the Binance Futures Testnet. The bot supports a variety of order types including Market, Limit, Stop-Limit, OCO, TWAP, and Grid orders. It offers an interactive CLI interface for retail traders, developers, or researchers to test and simulate algorithmic trading strategies in a sandbox environment with testnet credentials.

Key Features

1. Interactive CLI with guided prompts for each order type
2. Supports Market, Limit, Stop-Limit, OCO, TWAP, and Grid orders
3. Robust input validation for all user inputs
4. Secure handling of Binance API keys using environment variables
5. Detailed logging of all requests and responses to bot.log
6. Modular codebase for easier maintenance and scalability
7. Ready-to-use with Binance Futures Testnet

Tech Stack

1. Language: Python 3.10+
2. HTTP Requests: requests
3. Environment Variables: python-dotenv
4. CLI Interface: argparse, input()
5. Logging: Python built-in logging
6. Data Handling: Python standard libraries
7. External API: Binance Futures Testnet REST API

API Setup Instructions

1. Go to Binance Futures Testnet.
2. Register or log in to your testnet account.
3. Navigate to API Management and create a new API Key.
4. Store the generated API_KEY and API_SECRET.
5. **Create a .env file in the root directory** and add the following:

BINANCE_API_KEY=your_testnet_api_key_here

```
BINANCE_API_SECRET=your_testnet_secret_key_here
```

Note: If you're using the testnet environment, switch the BASE_URL to https://testnet.binancefuture.com if needed.

Project Folder Structure

```
Sagars_Binance_Bot/

│

├── .env                    # API keys and config

├── requirements.txt          # Python dependencies

├── bot.log                   # Log file with order actions

├── README.md                 # Project overview

│

├── src/

│   ├── cli.py                # Main CLI runner

│   ├── config.py             # API and base URL configs

│   ├── logger.py             # Logger setup

│

│   ├── market_orders.py      # Market order logic

│   ├── limit_orders.py       # Limit order logic

│

│   ├── advanced/

│   │   ├── stop_limit.py     # Stop-Limit orders

│   │   ├── oco.py            # OCO orders

│   │   ├── twap.py           # TWAP orders

│   │   ├── grid.py           # Grid strategy

│

│  └── utils/

│      └── validators.py      # Input validation functions
```

How to Run the Bot

1. Clone the repository:

```
git init
git clone https://github.com/sagar1024/Sagars_Binance_Bot.git
```

2. Navigate into the project folder:

```
bash
cd Sagars_Binance_Bot
```

3. Install the dependencies using pip:

```
pip install -r requirements.txt
```

4. Navigate to the src directory:

```
cd src
```

5. Run the bot using:

```
python cli.py
```

6. Follow the prompts to enter:

Order type (MARKET, LIMIT, etc.)

Trading pair (e.g., BTCUSDT)

Side (BUY or SELL)

Price, quantity, and other parameters as per order type

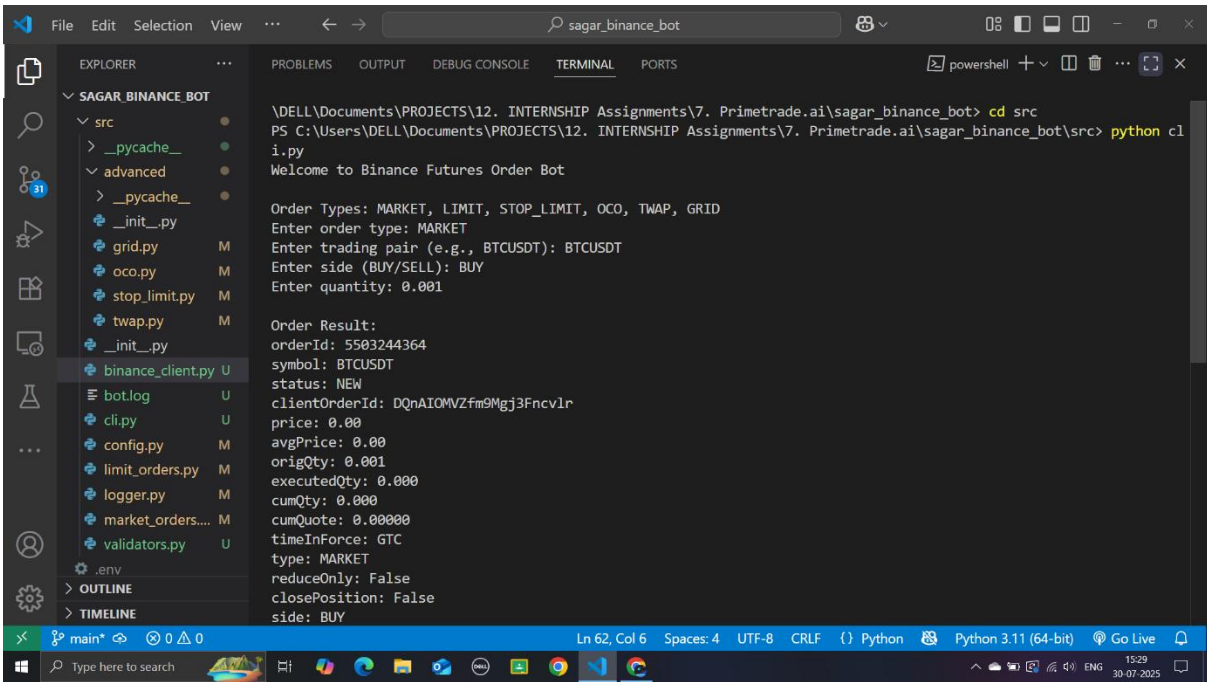
7. All logs from the session, including request and response payloads, will be stored in bot.log.

Developed by –

Sagar Gurung
Python Developer | AI Engineer | Blockchain & Trading Bots Enthusiast
Github Repository – https://github.com/sagar1024/Sagars_Binance_Bot

Output screenshots –

MARKET | BUY

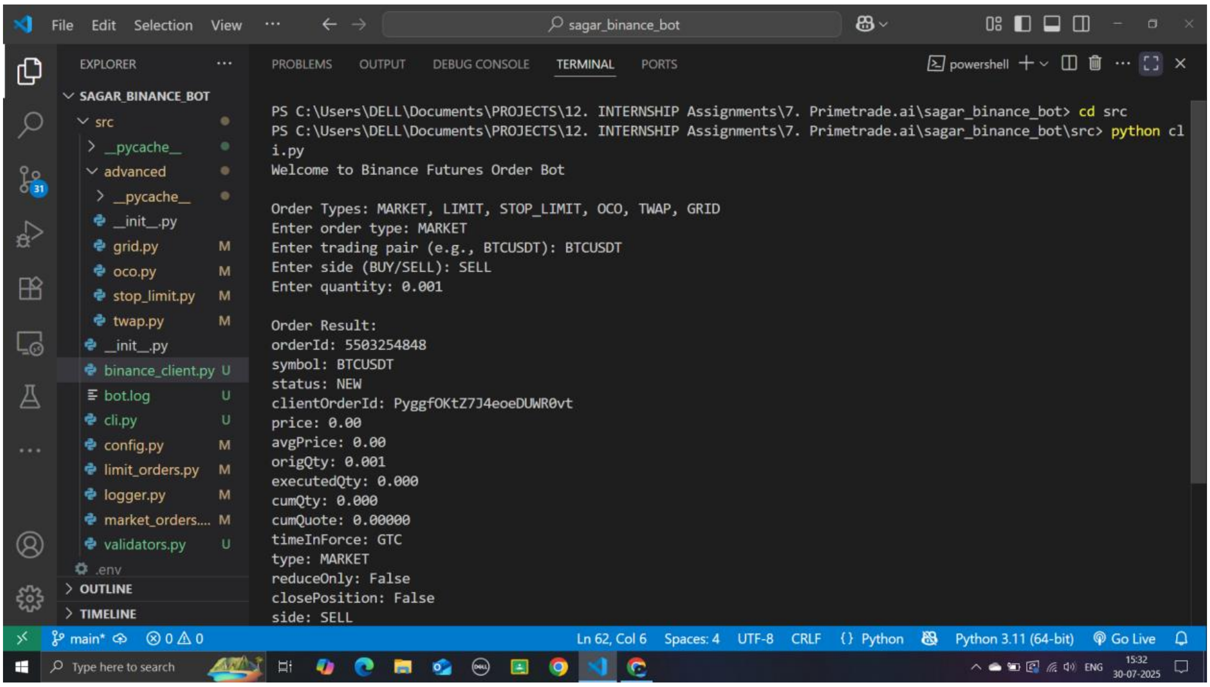


```
\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot> cd src
PS C:\Users\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot\src> python cl
i.py
Welcome to Binance Futures Order Bot

Order Types: MARKET, LIMIT, STOP_LIMIT, OCO, TWAP, GRID
Enter order type: MARKET
Enter trading pair (e.g., BTCUSDT): BTCUSDT
Enter side (BUY/SELL): BUY
Enter quantity: 0.001

Order Result:
orderId: 5503244364
symbol: BTCUSDT
status: NEW
clientOrderId: DQnAIOMVZfm9Mgj3Fncvlr
price: 0.00
avgPrice: 0.00
origQty: 0.001
executedQty: 0.000
cumQty: 0.000
cumQuote: 0.00000
timeInForce: GTC
type: MARKET
reduceOnly: False
closePosition: False
side: BUY
```

MARKET | SELL

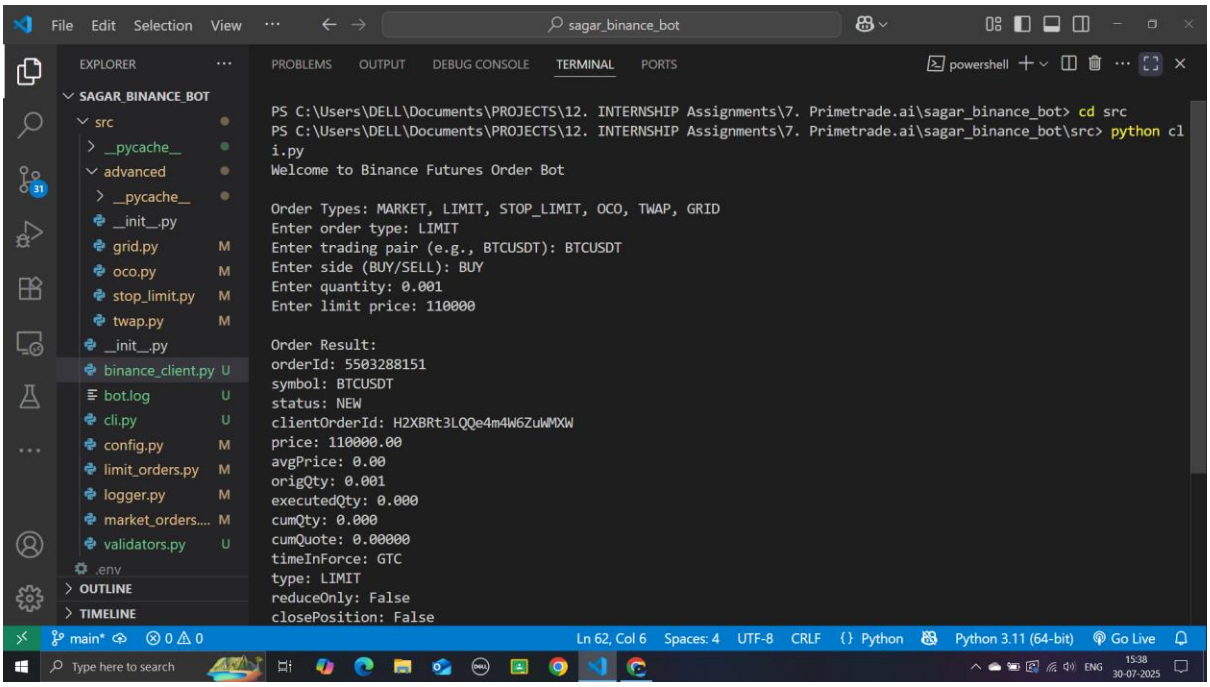


```
PS C:\Users\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot> cd src
PS C:\Users\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot\src> python cl
i.py
Welcome to Binance Futures Order Bot

Order Types: MARKET, LIMIT, STOP_LIMIT, OCO, TWAP, GRID
Enter order type: MARKET
Enter trading pair (e.g., BTCUSDT): BTCUSDT
Enter side (BUY/SELL): SELL
Enter quantity: 0.001

Order Result:
orderId: 5503254848
symbol: BTCUSDT
status: NEW
clientOrderId: PyggfOKtZ7J4eoeDUWR0vt
price: 0.00
avgPrice: 0.00
origQty: 0.001
executedQty: 0.000
cumQty: 0.000
cumQuote: 0.00000
timeInForce: GTC
type: MARKET
reduceOnly: False
closePosition: False
side: SELL
```

LIMIT | BUY

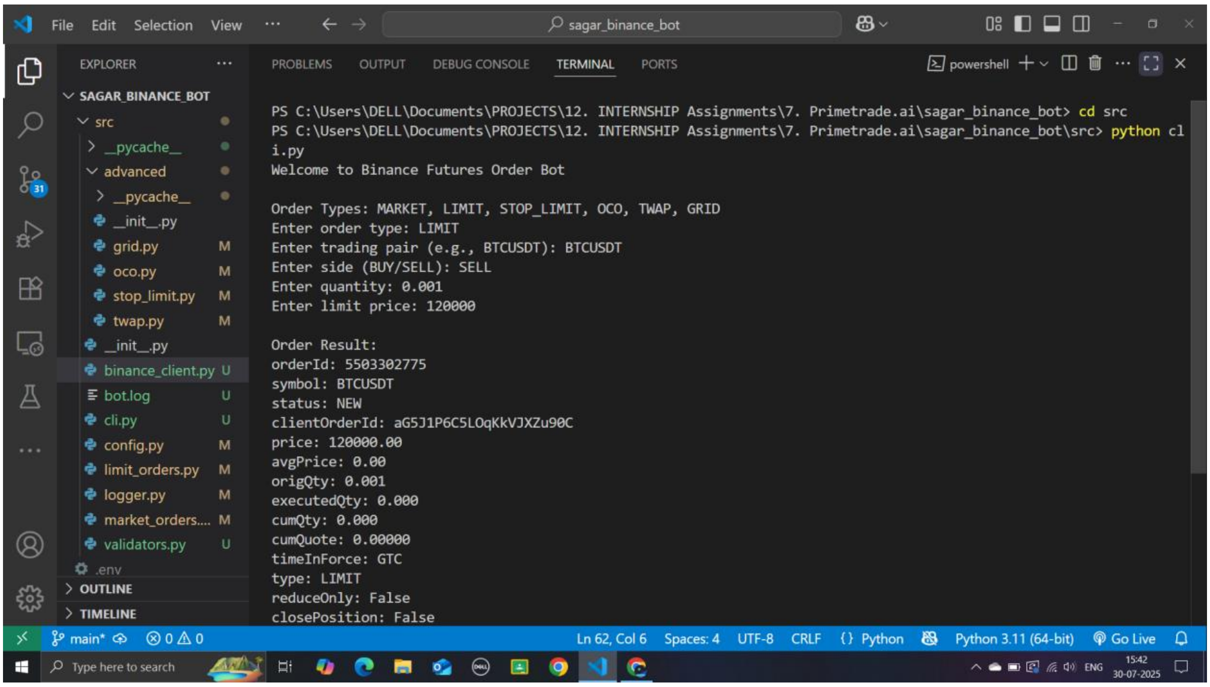


```
PS C:\Users\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot> cd src
PS C:\Users\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot\src> python cl
i.py
Welcome to Binance Futures Order Bot

Order Types: MARKET, LIMIT, STOP_LIMIT, OCO, TWAP, GRID
Enter order type: LIMIT
Enter trading pair (e.g., BTCUSDT): BTCUSDT
Enter side (BUY/SELL): BUY
Enter quantity: 0.001
Enter limit price: 110000

Order Result:
orderId: 5503288151
symbol: BTCUSDT
status: NEW
clientOrderId: H2XBRt3LQQe4m4W6ZuWMXW
price: 110000.00
avgPrice: 0.00
origQty: 0.001
executedQty: 0.000
cumQty: 0.000
cumQuote: 0.00000
timeInForce: GTC
type: LIMIT
reduceOnly: False
closePosition: False
```

LIMIT | SELL

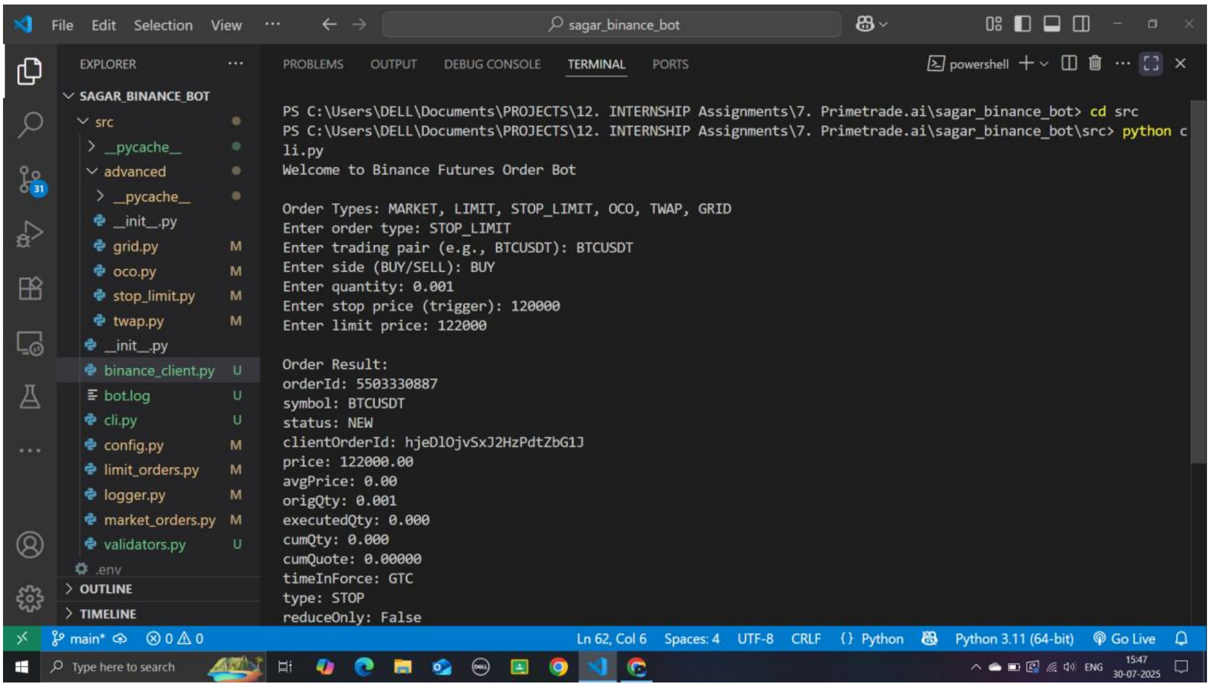


```
PS C:\Users\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot> cd src
PS C:\Users\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot\src> python cl
i.py
Welcome to Binance Futures Order Bot

Order Types: MARKET, LIMIT, STOP_LIMIT, OCO, TWAP, GRID
Enter order type: LIMIT
Enter trading pair (e.g., BTCUSDT): BTCUSDT
Enter side (BUY/SELL): SELL
Enter quantity: 0.001
Enter limit price: 120000

Order Result:
orderId: 5503302775
symbol: BTCUSDT
status: NEW
clientOrderId: aG5J1P6C5LOqKkVJXZu90C
price: 120000.00
avgPrice: 0.00
origQty: 0.001
executedQty: 0.000
cumQty: 0.000
cumQuote: 0.00000
timeInForce: GTC
type: LIMIT
reduceOnly: False
closePosition: False
```

STOP_LIMIT | BUY

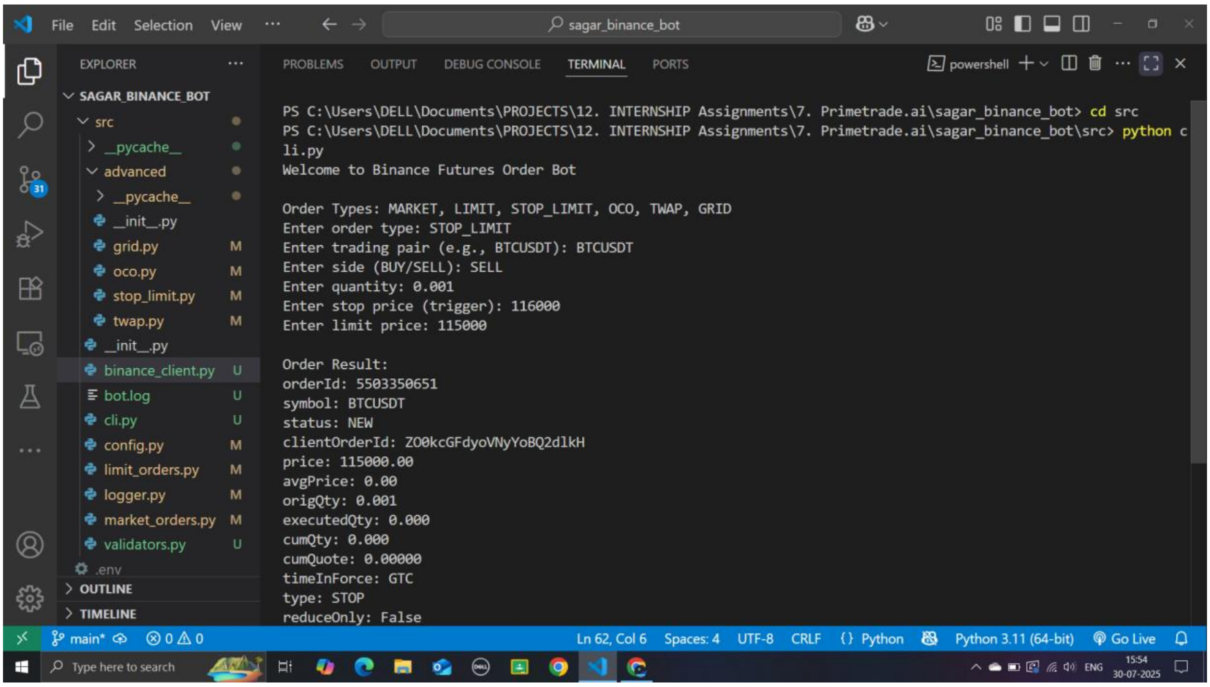


```
PS C:\Users\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot> cd src
PS C:\Users\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot\src> python c
li.py
Welcome to Binance Futures Order Bot

Order Types: MARKET, LIMIT, STOP_LIMIT, OCO, TWAP, GRID
Enter order type: STOP_LIMIT
Enter trading pair (e.g., BTCUSDT): BTCUSDT
Enter side (BUY/SELL): BUY
Enter quantity: 0.001
Enter stop price (trigger): 120000
Enter limit price: 122000

Order Result:
orderId: 5503330887
symbol: BTCUSDT
status: NEW
clientOrderId: hjeDlOjvSxJ2HzPdtZbG1J
price: 122000.00
avgPrice: 0.00
origQty: 0.001
executedQty: 0.000
cumQty: 0.000
cumQuote: 0.00000
timeInForce: GTC
type: STOP
reduceOnly: False
```

STOP_LIMIT | SELL

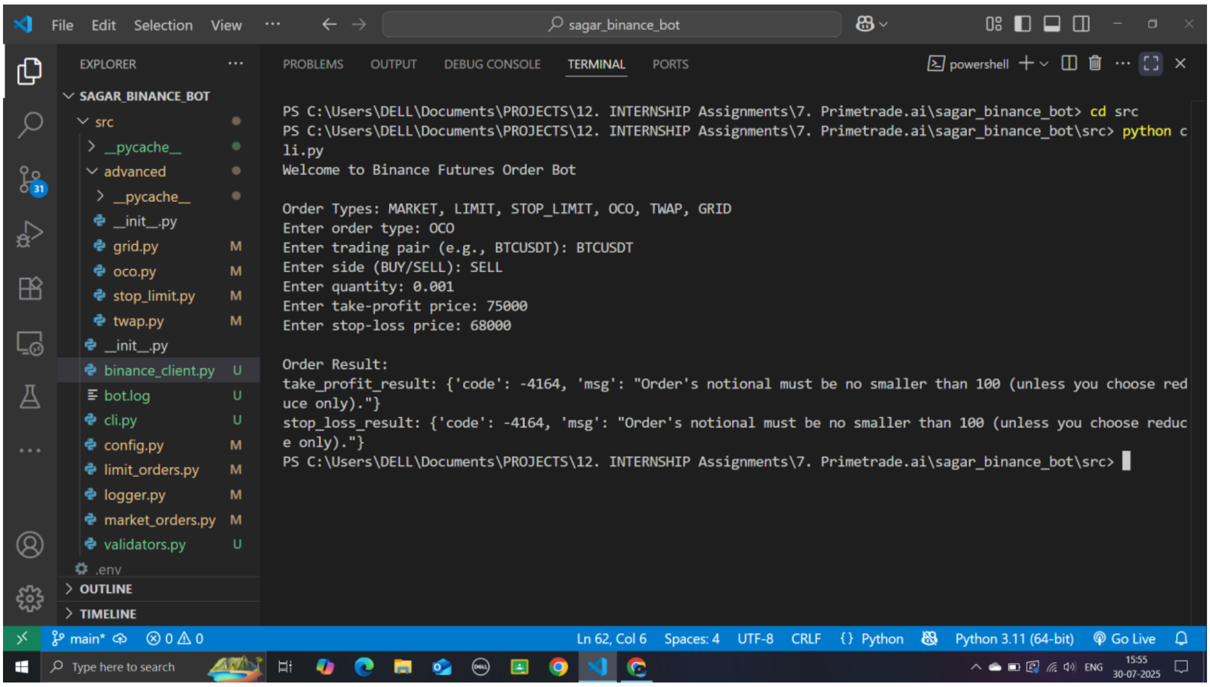


```
PS C:\Users\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot> cd src
PS C:\Users\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot\src> python c
li.py
Welcome to Binance Futures Order Bot

Order Types: MARKET, LIMIT, STOP_LIMIT, OCO, TWAP, GRID
Enter order type: STOP_LIMIT
Enter trading pair (e.g., BTCUSDT): BTCUSDT
Enter side (BUY/SELL): SELL
Enter quantity: 0.001
Enter stop price (trigger): 116000
Enter limit price: 115000

Order Result:
orderId: 5503350651
symbol: BTCUSDT
status: NEW
clientOrderId: ZO0kcGFdyoVNyYoBQ2dlkH
price: 115000.00
avgPrice: 0.00
origQty: 0.001
executedQty: 0.000
cumQty: 0.000
cumQuote: 0.00000
timeInForce: GTC
type: STOP
reduceOnly: False
```

OCO | SELL (SIMULATED)

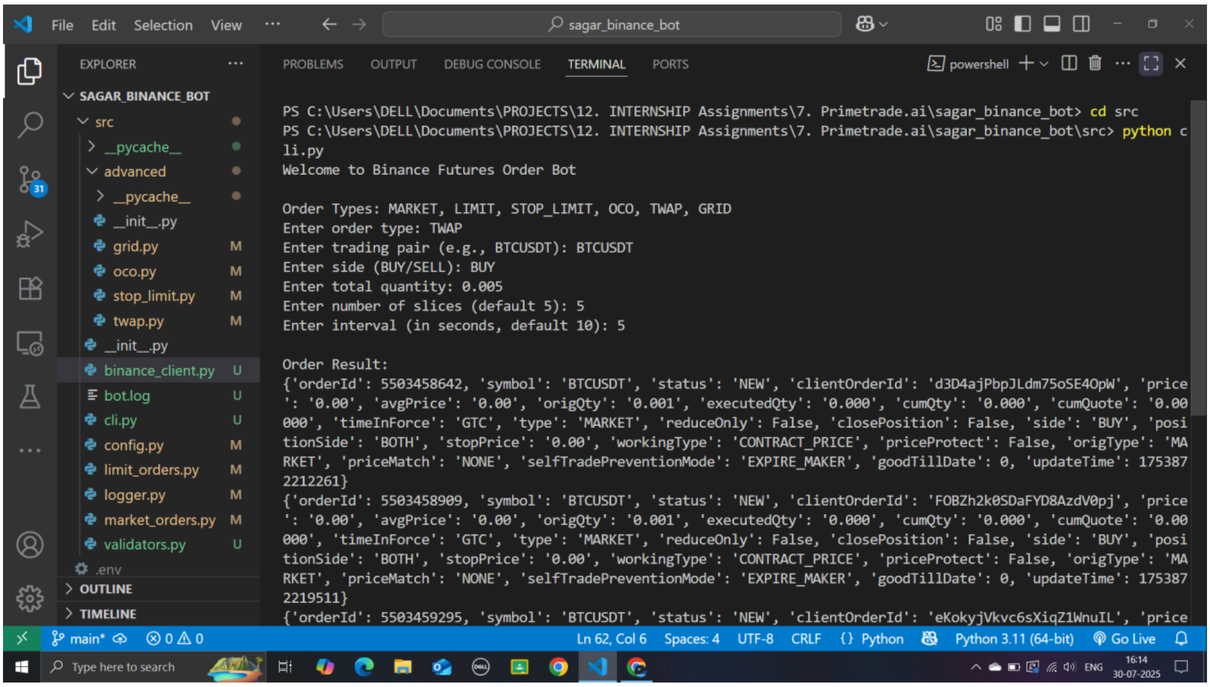


```
PS C:\Users\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot> cd src
PS C:\Users\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot\src> python c
li.py
Welcome to Binance Futures Order Bot

Order Types: MARKET, LIMIT, STOP_LIMIT, OCO, TWAP, GRID
Enter order type: OCO
Enter trading pair (e.g., BTCUSDT): BTCUSDT
Enter side (BUY/SELL): SELL
Enter quantity: 0.001
Enter take-profit price: 75000
Enter stop-loss price: 68000

Order Result:
take_profit_result: {'code': -4164, 'msg': "Order's notional must be no smaller than 100 (unless you choose red
uce only)."}
stop_loss_result: {'code': -4164, 'msg': "Order's notional must be no smaller than 100 (unless you choose reduc
e only)."}
PS C:\Users\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot\src>
```

TWAP | BUY

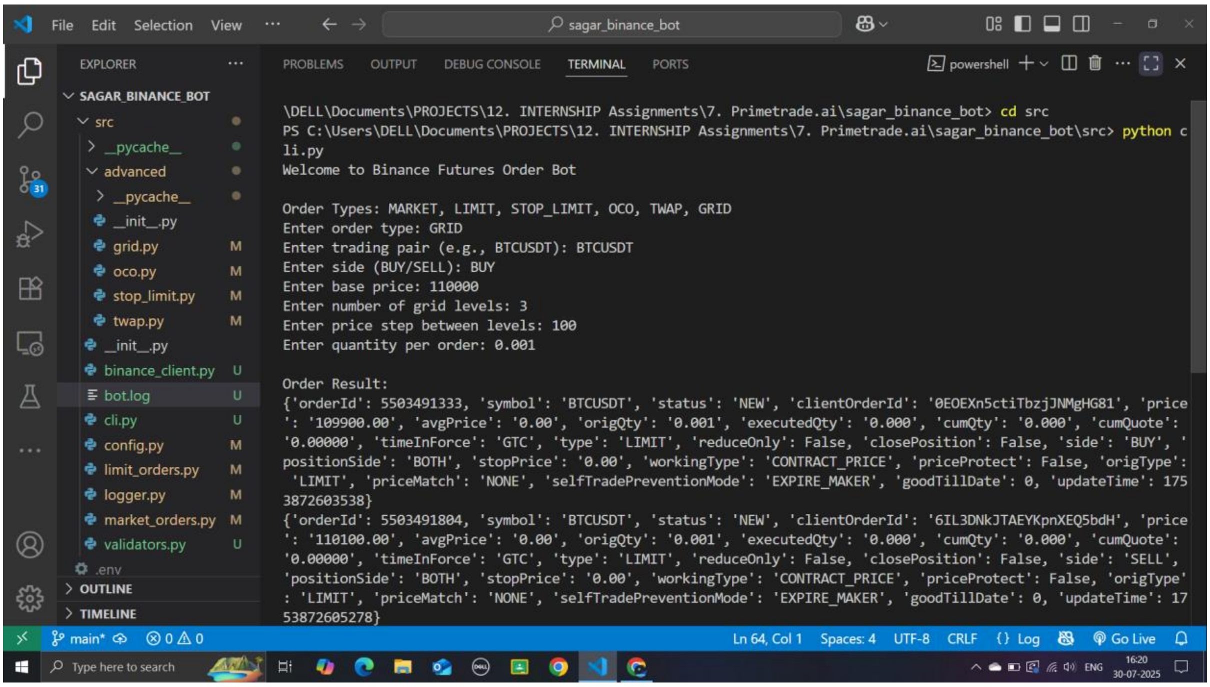


```
PS C:\Users\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot> cd src
PS C:\Users\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot\src> python c
li.py
Welcome to Binance Futures Order Bot

Order Types: MARKET, LIMIT, STOP_LIMIT, OCO, TWAP, GRID
Enter order type: TWAP
Enter trading pair (e.g., BTCUSDT): BTCUSDT
Enter side (BUY/SELL): BUY
Enter total quantity: 0.005
Enter number of slices (default 5): 5
Enter interval (in seconds, default 10): 5

Order Result:
{'orderId': 5503458642, 'symbol': 'BTCUSDT', 'status': 'NEW', 'clientOrderId': 'd3D4ajPbpJLdm75oSE4OpW', 'price
': '0.00', 'avgPrice': '0.00', 'origQty': '0.001', 'executedQty': '0.000', 'cumQty': '0.000', 'cumQuote': '0.00
000', 'timeInForce': 'GTC', 'type': 'MARKET', 'reduceOnly': False, 'closePosition': False, 'side': 'BUY', 'posi
tionSide': 'BOTH', 'stopPrice': '0.00', 'workingType': 'CONTRACT_PRICE', 'priceProtect': False, 'origType': 'MA
RKET', 'priceMatch': 'NONE', 'selfTradePreventionMode': 'EXPIRE_MAKER', 'goodTillDate': 0, 'updateTime': 175387
2212261}
{'orderId': 5503458909, 'symbol': 'BTCUSDT', 'status': 'NEW', 'clientOrderId': 'FOBZh2k0SDaFYD8AzdV0pj', 'price
': '0.00', 'avgPrice': '0.00', 'origQty': '0.001', 'executedQty': '0.000', 'cumQty': '0.000', 'cumQuote': '0.00
000', 'timeInForce': 'GTC', 'type': 'MARKET', 'reduceOnly': False, 'closePosition': False, 'side': 'BUY', 'posi
tionSide': 'BOTH', 'stopPrice': '0.00', 'workingType': 'CONTRACT_PRICE', 'priceProtect': False, 'origType': 'MA
RKET', 'priceMatch': 'NONE', 'selfTradePreventionMode': 'EXPIRE_MAKER', 'goodTillDate': 0, 'updateTime': 175387
2219511}
{'orderId': 5503459295, 'symbol': 'BTCUSDT', 'status': 'NEW', 'clientOrderId': 'eKokyjVkvc6sXiqZ1WnuIL', 'price
```

GRID | BUY



```
\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot> cd src
PS C:\Users\DELL\Documents\PROJECTS\12. INTERNSHIP Assignments\7. Primetrade.ai\sagar_binance_bot\src> python c
li.py
Welcome to Binance Futures Order Bot

Order Types: MARKET, LIMIT, STOP_LIMIT, OCO, TWAP, GRID
Enter order type: GRID
Enter trading pair (e.g., BTCUSDT): BTCUSDT
Enter side (BUY/SELL): BUY
Enter base price: 110000
Enter number of grid levels: 3
Enter price step between levels: 100
Enter quantity per order: 0.001

Order Result:
{'orderId': 5503491333, 'symbol': 'BTCUSDT', 'status': 'NEW', 'clientOrderId': '0EOEXn5ctiTbzjJNMgHG81', 'price
': '109900.00', 'avgPrice': '0.00', 'origQty': '0.001', 'executedQty': '0.000', 'cumQty': '0.000', 'cumQuote':
'0.00000', 'timeInForce': 'GTC', 'type': 'LIMIT', 'reduceOnly': False, 'closePosition': False, 'side': 'BUY', '
positionSide': 'BOTH', 'stopPrice': '0.00', 'workingType': 'CONTRACT_PRICE', 'priceProtect': False, 'origType':
 'LIMIT', 'priceMatch': 'NONE', 'selfTradePreventionMode': 'EXPIRE_MAKER', 'goodTillDate': 0, 'updateTime': 175
3872603538}
{'orderId': 5503491804, 'symbol': 'BTCUSDT', 'status': 'NEW', 'clientOrderId': '6IL3DNkJTAEYKpnXEQ5bdH', 'price
': '110100.00', 'avgPrice': '0.00', 'origQty': '0.001', 'executedQty': '0.000', 'cumQty': '0.000', 'cumQuote':
'0.00000', 'timeInForce': 'GTC', 'type': 'LIMIT', 'reduceOnly': False, 'closePosition': False, 'side': 'SELL',
'positionSide': 'BOTH', 'stopPrice': '0.00', 'workingType': 'CONTRACT_PRICE', 'priceProtect': False, 'origType'
: 'LIMIT', 'priceMatch': 'NONE', 'selfTradePreventionMode': 'EXPIRE_MAKER', 'goodTillDate': 0, 'updateTime': 17
53872605278}
```