

AI Engineering Intern Assignment: Build a Strategy Agent

Objective:

Create an **AI-powered agent** that takes in natural language strategy rules for backtesting and generates a Python script to run the strategy using historical data fetched from a custom data module.

Structure:

You will create two Python files:

1. `data_engine.py`
 2. `agent.py`
-

1. `data_engine.py`

Implement a function:

```
def fetch_historicaldata(stock, from_date, to_date, timeframe):
```

- This function must return a **Pandas DataFrame** with the following columns:
 - `stock, date, timestamp, open, high, low, close, volume`
- You can generate **dummy data** using `pandas` and `numpy`.
- The `stock` value can be hardcoded to `"NIFTY"`.

Example:

```
# Sample Output:
```

```
# stock | date          | timestamp | open | high | low | close | volume
# NIFTY | 2023-01-01 | 09:15    | 100  | 110  | 95  | 105   | 10000
```

2. agent.py

Build a Python script that does the following:

- Takes **user input** describing a strategy rule in plain English (e.g., "Buy when close > open").
- Automatically:
 1. Uses the `fetch_historicaldata()` function from `data_engine.py`.
 2. Converts the user-defined rule into executable Python code.
 3. Generates a complete **Python backtest script** file (`generated_strategy.py`) that:
 - Loads the historical data using the function
 - Applies the strategy logic
 - Prints entry/exit signals or results

Example Flow:

```
$ python agent.py
Enter your strategy rule: Buy when close > open
```

Output:

```
✓ Strategy saved in generated_strategy.py
```

Tips:

- Use **Python code generation techniques** (e.g., string formatting or `exec`).

- Make sure `agent.py` imports and uses `fetch_historicaldata()` from `data_engine`.
 - Assume all strategies will be simple comparisons (`close > open`, `high > close`, etc.).
-

Submission:

Submit a zip file or GitHub repo containing:

- `data_engine.py`
 - `agent.py`
 - Any dependencies in `requirements.txt` (if needed)
 - Sample output of the generated strategy file (e.g., `generated_strategy.py`)
-