REPORT

Explanation of the Python File Attached from Lines 2 through 30 and 32 through 36

Lines 2 - 4
class PharmacyStock: This defines the blueprint for your system.
__init__: This is the constructor. It runs automatically when you create a new instance (like on line 28).
self.stock = {}: This initializes an empty Python dictionary. Think of this as an empty shelf or a blank ledger where all the medicine data will be stored.

Lines 5 - 9

- **The Check:** First, it checks `if name in self.stock`. You don't want to overwrite existing data accidentally.
- **The Action:** If the name is new, it creates a nested dictionary: `'Aspirin': {'quantity': 100, 'price': 5.99}`.
- **The Return:** It sends back a text message confirming the action.

Lines 10 - 13
The Check: It verifies if the medicine exists. If not, it returns an error message.
The Action: If found, it returns the dictionary containing that medicine's quantity and price

Lines 14 - 21
The Flexibility: Notice `quantity=None` and `price=None` in the definition. This makes these arguments **optional**.
The Logic:

- If you provide a new `quantity`, it updates the quantity.
- If you provide a new `price`, it updates the price.
- If you provide *both*, it updates *both*.
- This allows you to change the price without accidentally deleting the quantity count.

Lines 22 - 26

The Action: The `del` keyword permanently removes the key (the medicine name) and its value (the details) from the dictionary.

Lines 27 - 32
This section actually uses the class. It creates an instance called `pharmacy` and tests every function to prove they work, printing the results to the console so the user can see the "Success" or "Error" messages returned by the methods.