Problem Title: Supply Chain Inventory Tracker

Problem Statement

You are building a **Supply Chain Inventory Tracker** for a retail chain. Each product in the supply chain has:

- **Key** → **Product ID** (`String`, unique)
- **Value** → **Units in stock** (`Integer`)

The system manages the inflow and outflow of goods and offers utilities to manage restocking and listing items in stock.

Class Declaration

```
class SupplyChainInventory:
```

The `__init__` Method

```
def __init__(self):
    self.stock = {}
```

Initializes an empty dictionary to maintain the product stock.

Operations (Methods)

1. Add Product Stock

```
def add_product(self, product_id: str, quantity: int) -> dict:
```

If the product already exists, increase its stock.
Otherwise, create a new entry.
Return the updated stock dictionary.

---

2. Fulfill Order

```python
def fulfill_order(self, product_id: str, quantity: int) -> dict:
```

If sufficient quantity exists, deduct from stock.
If not, raise a `ValueError`.
Return the updated stock dictionary.

---

3. Restock Return

```python
def restock_return(self, product_id: str, quantity: int) -> dict:
```

Add returned quantity back into stock.
If product doesn't exist, add it.
Return the updated stock dictionary.

---

4. List Available Stock

```python
def list_available_products(self) -> list:
```

Return list of `product_id`s with stock > 0.

Example Behavior

```python
sc = SupplyChainInventory()
sc.add_product("PRD101", 50)        # {'PRD101': 50}
sc.fulfill_order("PRD101", 10)      # {'PRD101': 40}
sc.restock_return("PRD101", 5)      # {'PRD101': 45}
sc.list_available_products()        # ['PRD101']
```

☑ Visible Test Cases

TC#	Description	Function
TC1	Adding a new product	`add_product()`
TC2	Fulfilling an order	`fulfill_order()`
TC3	Restocking returned goods	`restock_return()`

Hidden Test Cases

HTC#	Description	Function
HTC1	Listing only products with stock >0	`list_available_products()`
HTC2	Order with insufficient stock	`fulfill_order()` — should raise `ValueError`

- `run.py`: Script to run visible test cases locally.

- `requirements.txt`: Lists dependencies (if any).