ARMY INVENTORY MANAGEMENT SYSTEM

Submitted By: **Kushagra Singh Rajwar**
Registration number-25BEC10052
B.Tech 1st Year

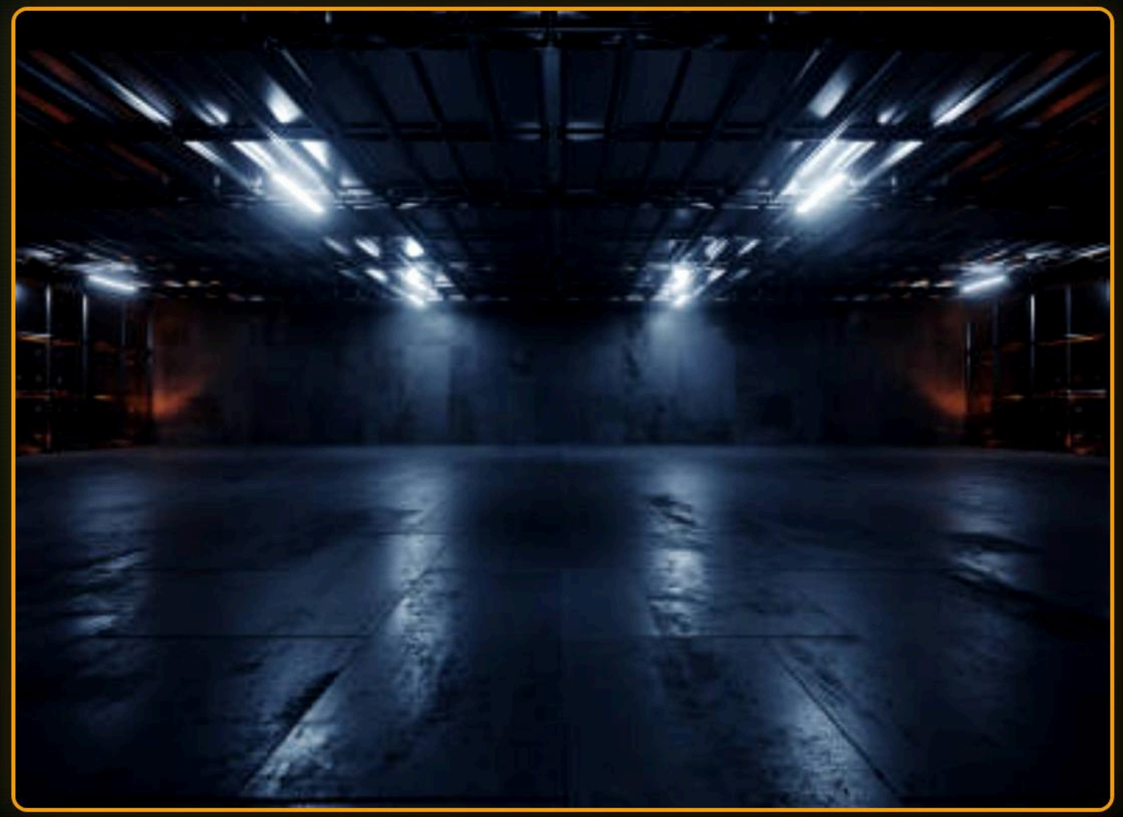
1. INTRODUCTION

MISSION OVERVIEW

The Army Inventory Management System is a command-line tool developed to simulate military logistics.

It acts as a central dashboard for a Base Commander to:

- View real-time stock levels across bases.
- Calculate total asset value instantly.
- Mobilize supplies between camps efficiently.



2. OBJECTIVES

AUTOMATE

Replace manual paper records with a digital system to reduce human error and increase retrieval speed.

APPLY PYTHON

Demonstrate mastery of core concepts like Nested Dictionaries, Control Flow loops, and Input Validation.

SIMULATE LOGIC

Model real-world constraints, such as ensuring a base cannot transfer items it does not physically possess.

3. PROBLEM STATEMENT

CURRENT SCENARIO

Traditional methods involve disconnected spreadsheets or paper logs.

- Slow information retrieval.
- High risk of calculation errors.
- No global visibility of assets.

PROPOSED SOLUTION

A centralized Python Command Interface.

- Instant global search.
- Automatic budget calculation.
- Dynamic camp creation.

4. TECHNOLOGY USED

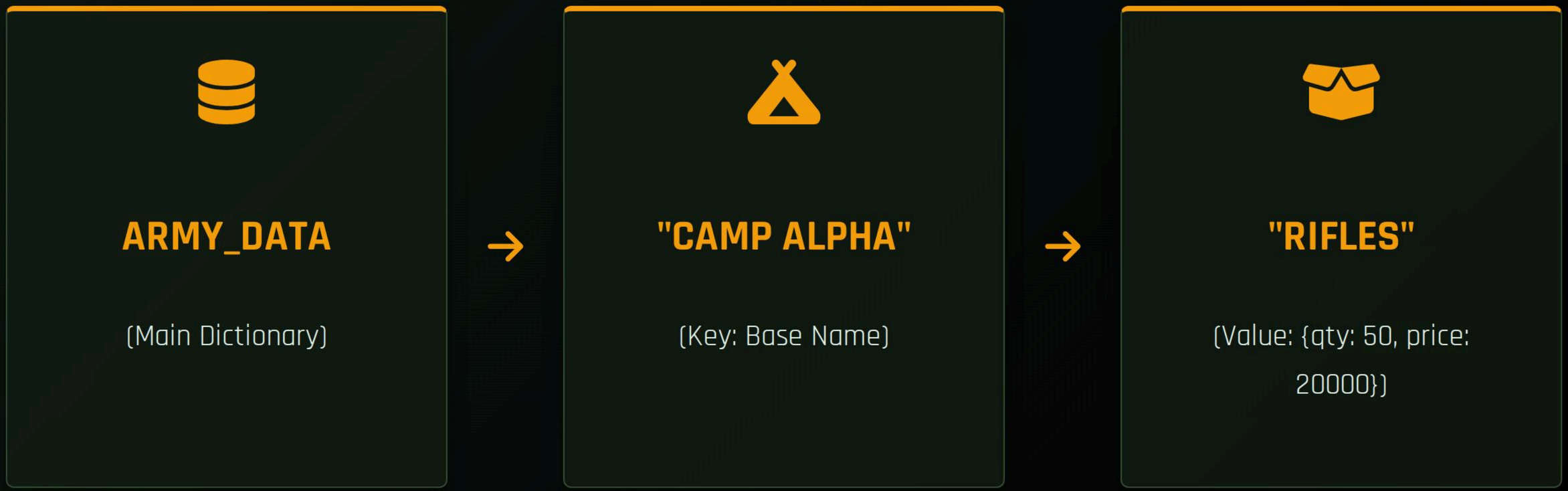
CORE STACK

The system is built entirely using Python 3.x, focusing on efficient data structures.

Component	Usage
Dictionaries	Nested storage for Camps & Items.
While Loops	Main menu dashboard loop.
Input/Output	User interaction & validation.

5. SYSTEM DESIGN (DATA STRUCTURE)

The system uses a **Nested Dictionary** structure to organize data hierarchically.



6. IMPLEMENTATION DETAILS

GLOBAL SEARCH

Iterates through every camp in the database to find a specific item (e.g., "Grenades"). It aggregates the count from all bases to provide a total available quantity.

LOGISTICS TRANSFER

Validates if the source camp has enough stock. If valid, it subtracts the qty from the source and adds it to the destination, creating new records if necessary.

```python
import time

def main():
    # Initial data
    army_data = {
        "Camp Alpha": {
            "Rifles": {"qty": 50, "price": 20000},
            "Rations": {"qty": 1000, "price": 50}
        },
        "Camp Bravo": {
            "Rifles": {"qty": 10, "price": 20000},
            "Medkits": {"qty": 20, "price": 500}
        }
    }

    while True:
        print("\n=== IN COMMAND HQ DASHBOARD ===")
        print("1. View All Camps & Assets")
        print("2. Add New Camp / Update Stock")
        print("3. SEARCH Item (Global Intel)")
        print("4. TRANSFER Supplies (Logistics)")
        print("5. Exit")

        choice = input("Enter Command: ")

        # VIEW REPORT
        if choice == '1':
            print("\n--- SITUATION REPORT ---")
            grand_total = 0
            for camp, items in army_data.items():
                print(f" {camp}")
                for item, details in items.items():
                    val = details['qty'] * details['price']
                    print(f"   - {item}: {details['qty']} units | Val: ₹{val}")
                    grand_total += val
            print(f"\n TOTAL MILITARY BUDGET DEPLOYED: ₹{grand_total}")

        # 2. ADD DATA
        elif choice == '2':
            camp = input("Enter Camp Name: ").strip()
            if camp not in army_data:
                print(f"Creating new base: {camp}...")
                army_data[camp] = {}

            item = input("Enter Item Name: ").title()
            try:
                qty = int(input("Enter Quantity: "))
                price = int(input("Enter Price (₹): "))
                army_data[camp][item] = {"qty": qty, "price": price}
                print(" Inventory Updated.")
            except ValueError:
                print(" Error: Use numbers for quantity/price.")

        elif choice == '3':
            target = input("Enter Item to find (e.g. Rifles): ").title()
            print(f"\n SEARCHING FOR: {target}...")
            found = False
            total_count = 0

            for camp, items in army_data.items():
                if target in items:
                    count = items[target]['qty']
                    print(f"   -> Found at {camp}: {count} units")
                    total_count += count
                    found = True

            if found:
                print(f" Total {target} across all camps: {total_count}")
            else:
                print(" Item not found in any camp.")

        elif choice == '4':
            print("\n--- LOGISTICS TRANSFER ---")
            source = input("From Camp: ").strip()
            dest = input("To Camp: ").strip()
            item = input("Item to move: ").title()

            if source in army_data and dest in army_data:

                if item in army_data[source]:
                    current_qty = army_data[source][item]['qty']
                    print(f"Available at {source}: {current_qty}")

                    try:
                        move_qty = int(input("Quantity to move: "))


                        if move_qty <= current_qty:

                            army_data[source][item]['qty'] -= move_qty


                            if item not in army_data[dest]:

                                price = army_data[source][item]['price']
                                army_data[dest][item] = {"qty": 0, "price": price}

                            army_data[dest][item]['qty'] += move_qty

                            print(f" TRANSFER COMPLETE: Moved {move_qty} {item} from {source} to {dest}.")
                        else:
                            print(" Insufficient stock!")
                    except ValueError:
                        print(" Invalid number.")
                else:
                    print(f" {source} does not have {item}.")
            else:
                print("One or both camps do not exist.")

        elif choice == '5':
            print("Jai Hind. Over And Out!")
            break
        else:
            print("Invalid Command.")

if __name__ == "__main__":
    main()
```

7. OUTPUT & RESULTS

> TERMINAL OUTPUT 1

📍 Camp Alpha
- Rifles: 50 units | Val: ₹10,00,000
- Rations: 1000 units | Val: ₹50,000

💰 CAMP VALUE: ₹10,50,000

> TERMINAL OUTPUT 2

From Camp: Camp Alpha
To Camp: Camp Bravo
Item: Rifles
Quantity: 5

✅ TRANSFER COMPLETE: Moved 5 Rifles from Camp Alpha to Camp Bravo.

```
=== IN COMMAND HQ DASHBOARD ===
1. View All Camps & Assets
2. Add New Camp / Update Stock
3. SEARCH Item (Global Intel)
4. TRANSFER Supplies (Logistics)
5. Exit
Enter Command: 2
Enter Camp Name: Aplha
Creating new base: Aplha...
Enter Item Name: Rifle
Enter Quantity: 300
Enter Price (₹): 10000000
 Inventory Updated.

=== IN COMMAND HQ DASHBOARD ===
1. View All Camps & Assets
2. Add New Camp / Update Stock
3. SEARCH Item (Global Intel)
4. TRANSFER Supplies (Logistics)
5. Exit
Enter Command: 1

--- SITUATION REPORT ---
 Camp Alpha
   - Rifles: 50 units | Val: ₹1000000
   - Rations: 1000 units | Val: ₹50000
 Camp Bravo
```

8. CHALLENGES FACED

- **Nested Data Access:** Initial difficulty in accessing keys deep within the dictionary structure.
- **Logic Validation:** Creating the safety check to prevent users from transferring more items than physically available.
- **Data Persistence:** Understanding that data is currently reset when the program closes, highlighting the need for file handling.

9. FUTURE SCOPE

FILE HANDLING

Implement .txt or .csv saving to ensure data persists after the program closes.

SECURITY

Add a password authentication system for the Commander to access the dashboard.

GUI

Upgrade from CLI to a Graphical User Interface using the Tkinter library.

MISSION ACCOMPLISHED

The Army Inventory Management System successfully demonstrates the power of Python in Military Logistics.

JAI HIND