```
#include <iostream>
#include <vector>
#include <string>
#include <iomanip>
#include <algorithm>
#include <limits>

using namespace std;

const string RED = "\033[31m";
const string GREEN = "\033[32m";
const string RESET = "\033[0m";

struct Date {
    int day, month, year;
    Date(int d = 0, int m = 0, int y = 0) : day(d), month(m), year(y) {}

    friend ostream& operator<<(ostream& os, const Date& date) {
        os << setfill('0') << setw(2) << date.day << "/"
           << setfill('0') << setw(2) << date.month << "/"
           << date.year;
        return os;
    }
};

struct Item {
    string name;
    int quantity;
    double price;
    Date manufacturingDate;
    Date expiryDate;
```

```
    Item(string n, int q, double p, Date m, Date e) : name(n), quantity(q), price(p),
manufacturingDate(m), expiryDate(e) {}
};

class Inventory {
private:
    vector<Item> items;

public:
    void addItem(const Item& item) {
        items.push_back(item);
    }

    void deleteItem(const string& name) {
        items.erase(remove_if(items.begin(), items.end(), [&](const Item& item) { return item.name ==
name; }), items.end());
    }

    void displayInventory() {
        cout << GREEN << left << setw(20) << "Name" << setw(10) << "Quantity" << setw(10) << "Price"
<< setw(20) << "Manufacturing Date" << "Expiry Date" << RESET << "\n";
        cout << string(70, '-') << "\n";
        for (const auto& item : items) {
            cout << left << setw(20) << item.name
                 << setw(10) << item.quantity
                 << setw(10) << "$" << item.price
                 << setw(20) << item.manufacturingDate
                 << item.expiryDate << "\n";
        }
    }
```

```
    Item* searchItem(const string& name) {
        for (auto& item : items) {
            if (item.name == name) {
                return &item;
            }
        }
        return nullptr;
    }

    bool updateItem(const string& name, int quantity, double price) {
        Item* item = searchItem(name);
        if (!item) return false;
        item->quantity = quantity;
        item->price = price;
        return true;
    }
};

void displayMenu() {
    cout << GREEN << "\nInventory Management System\n" << RESET;
    cout << "1. Add item\n";
    cout << "2. Delete item\n";
    cout << "3. Display inventory\n";
    cout << "4. Search item\n";
    cout << "5. Update item\n";
    cout << "6. Exit\n";
    cout << "Enter your choice: ";
}

int main() {
    Inventory inventory;
```

```
int choice;

do {
   displayMenu();
   cin >> choice;
   cin.ignore(numeric_limits<streamsize>::max(), '\n');

   switch (choice) {
      case 1: {
         string name;
         int quantity;
         double price;
         Date mfgDate, expDate;

         cout << "Enter item name: ";
         getline(cin, name);

         cout << "Enter quantity: ";
         cin >> quantity;

         cout << "Enter price: ";
         cin >> price;

         cout << "Enter manufacturing date (DD MM YYYY): ";
         cin >> mfgDate.day >> mfgDate.month >> mfgDate.year;

         cout << "Enter expiry date (DD MM YYYY): ";
         cin >> expDate.day >> expDate.month >> expDate.year;

         Item newItem(name, quantity, price, mfgDate, expDate);
         inventory.addItem(newItem);
```

```
    cout << GREEN << "Item added successfully!" << RESET << "\n";
    break;
}
case 2: {
    string name;
    cout << "Enter the name of the item to delete: ";
    getline(cin, name);

    inventory.deleteItem(name);
    cout << GREEN << "Item deleted successfully!" << RESET << "\n";
    break;
}
case 3:
    inventory.displayInventory();
    break;
case 4: {
    string name;
    cout << "Enter item name to search: ";
    getline(cin, name);

    Item* item = inventory.searchItem(name);
    if (item) {
        cout << GREEN << "Name: " << item->name
             << ", Quantity: " << item->quantity
             << ", Price: $" << item->price
             << ", Manufacturing Date: " << item->manufacturingDate
             << ", Expiry Date: " << item->expiryDate << RESET << "\n";
    } else {
        cout << RED << "Item not found." << RESET << "\n";
    }
    break;
```

```
        }
        case 5: {
            string name;
            int quantity;
            double price;

            cout << "Enter item name to update: ";
            getline(cin, name);

            cout << "Enter new quantity: ";
            cin >> quantity;

            cout << "Enter new price: ";
            cin >> price;

            if (inventory.updateItem(name, quantity, price)) {
                cout << GREEN << "Item updated successfully!" << RESET << "\n";
            } else {
                cout << RED << "Item not found." << RESET << "\n";
            }

            break;
        }
        case 6:
            cout << GREEN << "Exiting...\n" << RESET;
            break;
        default:
            cout << RED << "Invalid choice. Please try again." << RESET << "\n";
    }
} while (choice != 6);
```

```
    return 0;
}
```