Store management system
Store management system is based on a concept of managing stores.
The user adds the details of the items like item code, item name, quantity, price per piece, Manufactured date, Expiry date. The system which includes CRUD operations. Search item details by item code or item name which includes detail of item code, item name, expiry date, number of quantity available.
Features:
- **Add items**
- **Remove items**
- **Update items**
- **View items**

```
#include <stdio.h>
#include <string.h>

// Structure to represent an item
struct Item {
   int itemCode;
   char itemName[50];
   int quantity;
   float price;
   char manufacturedDate[20];
   char expiryDate[20];
};

// Function to display item details
void displayItem(struct Item item) {
   printf("Item Code: %d\nItem Name: %s\nExpiry Date: %s\nQuantity Available: %d\n",
         item.itemCode, item.itemName, item.expiryDate, item.quantity);
}

// Function to search for an item by item code
int searchByCode(struct Item store[], int numItems, int itemCode) {
   for (int i = 0; i < numItems; i++) {
      if (store[i].itemCode == itemCode) {
         return i; // Return index of found item
      }
   }
   return -1; // Item not found
}

// Function to search for an item by item name
int searchByName(struct Item store[], int numItems, const char itemName[]) {
   for (int i = 0; i < numItems; i++) {
      if (strcmp(store[i].itemName, itemName) == 0) {
         return i; // Return index of found item
      }
   }
```

```
    return -1; // Item not found
}

int main() {
    struct Item store[100]; // Assuming there are at most 100 items in the store
    int numItems = 0;
    int choice;

    do {
        printf("\nStore Management System\n");
        printf("1. Add Item\n");
        printf("2. Search Item\n");
        printf("3. Remove Quantity\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1: // Add Item
                if (numItems < 100) {
                    struct Item newItem;

                    printf("Enter item code: ");
                    scanf("%d", &newItem.itemCode);
                    printf("Enter item name: ");
                    scanf("%s", newItem.itemName);
                    printf("Enter quantity: ");
                    scanf("%d", &newItem.quantity);
                    printf("Enter price per piece: ");
                    scanf("%f", &newItem.price);
                    printf("Enter manufactured date: ");
                    scanf("%s", newItem.manufacturedDate);
                    printf("Enter expiry date: ");
                    scanf("%s", newItem.expiryDate);

                    store[numItems] = newItem;
                    numItems++;
                    printf("Item added successfully.\n");
                } else {
                    printf("Store is full. Cannot add more items.\n");
                }
                break;

            case 2: // Search Item
                {
                    int searchChoice;
                    printf("Search by:\n1. Item Code\n2. Item Name\nEnter choice: ");
                    scanf("%d", &searchChoice);
```

```
        if (searchChoice == 1) {
            int itemCode;
            printf("Enter item code: ");
            scanf("%d", &itemCode);

            int index = searchByCode(store, numItems, itemCode);
            if (index != -1) {
                displayItem(store[index]);
            } else {
                printf("Item not found.\n");
            }
        } else if (searchChoice == 2) {
            char itemName[50];
            printf("Enter item name: ");
            scanf("%s", itemName);

            int index = searchByName(store, numItems, itemName);
            if (index != -1) {
                displayItem(store[index]);
            } else {
                printf("Item not found.\n");
            }
        }/* else {
            printf("Invalid choice.\n");
        }*/
    }
    break;

case 3: // Remove Quantity
    {
        int itemCode, quantityToRemove;
        printf("Enter item code: ");
        scanf("%d", &itemCode);

        int index = searchByCode(store, numItems, itemCode);
        if (index != -1) {
            displayItem(store[index]);

            printf("Enter quantity to remove: ");
            scanf("%d", &quantityToRemove);

            if (quantityToRemove <= store[index].quantity) {
                store[index].quantity -= quantityToRemove;
                printf("%d quantity of %s removed.\n", quantityToRemove, store[index].itemName);
            } else {
                printf("Insufficient quantity available.\n");
            }
```

```
            } else {
                printf("Item not found.\n");
            }
        }
        break;

    case 4: // Exit
        printf("Exiting...\n");
        break;

    default:
        printf("Invalid choice.\n");
  }
 } while (choice != 4);

 return 0;
}
```

```
//OUTPUT
Store Management System
1. Add Item
2. Search Item
3. Remove Quantity
4. Exit
Enter your choice: 1
Enter item code: 1
Enter item name: c
Enter quantity: 10
Enter price per piece: 10
Enter manufactured date: 2/2/2
Enter expiry date: 4/2/4
Item added successfully.

Store Management System
1. Add Item
2. Search Item
3. Remove Quantity
4. Exit
Enter your choice: 2
Search by:
1. Item Code
2. Item Name
Enter choice: 1
Enter item code: 1
Item Code: 1
Item Name: c
```

Expiry Date: 4/2/4
Quantity Available: 10

Store Management System
1. Add Item
2. Search Item
3. Remove Quantity
4. Exit
Enter your choice: 3
Enter item code: 1
Item Code: 1
Item Name: c
Expiry Date: 4/2/4
Quantity Available: 10
Enter quantity to remove: 4
4 quantity of c removed.

Store Management System
1. Add Item
2. Search Item
3. Remove Quantity
4. Exit
Enter your choice: 2
Search by:
1. Item Code
2. Item Name
Enter choice: 1
Enter item code: 1
Item Code: 1
Item Name: c
Expiry Date: 4/2/4
Quantity Available: 6

Store Management System
1. Add Item
2. Search Item
3. Remove Quantity
4. Exit
Enter your choice: 4
Exiting...

//main trial

```
#include <stdio.h>
#include <string.h>

// Structure to represent an item
struct Item {
   int itemCode;
   char itemName[50];
   int quantity;
   float price;
   char manufacturedDate[20];
   char expiryDate[20];

};

// Function to display item details
void displayItem(struct Item item) {
   printf("Item Code: %d\nItem Name: %s\nExpiry Date: %s\nQuantity Available: %d\n",
         item.itemCode, item.itemName, item.expiryDate, item.quantity);
}

// Function to search for an item by item code
int searchByCode(struct Item store[], int numItems, int itemCode) {
   for (int i = 0; i < numItems; i++) {
      if (store[i].itemCode == itemCode) {
         return i; // Return index of found item
      }
   }
   return -1; // Item not found
}

// Function to search for an item by item name
int searchByName(struct Item store[], int numItems, const char itemName[]) {
   for (int i = 0; i < numItems; i++) {
      if (strcmp(store[i].itemName, itemName) == 0) {
         return i; // Return index of found item
      }
   }
   return -1; // Item not found
}
void removeExpiredItems(struct Item store[], int *numItems, const char currentDate[]) {
   int removedCount = 0;

   for (int i = 0; i < *numItems; i++) {
```

```
        if (strcmp(store[i].expiryDate, currentDate) <= 0) {
            printf("Removing %d quantity of expired %s\n", store[i].quantity, store[i].itemName);
            removedCount++;
            // Shift items to remove expired item
            for (int j = i; j < *numItems - 1; j++) {
                store[j] = store[j + 1];
            }
            i--; // Decrement i to re-check the current position
            (*numItems)--;
        }
    }

    if (removedCount == 0) {
        printf("No expired items found.\n");
    }
}
void updateItem(struct Item store[], int count, int updateCode) {
    for (int i = 0; i < count; i++) {
        if (store[i].itemCode == updateCode) {
            printf("Enter updated quantity: ");
            scanf("%d", &store[i].quantity);
            printf("Enter updated price per piece: ");
            scanf("%f", &store[i].price);
            printf("Item with code %d updated successfully!\n", updateCode);
            return;
        }
    }
    printf("Item with code %d not found.\n", updateCode);
}

void viewItems(struct Item store[], int count) {
    printf("Item Details:\n");
    for (int i = 0; i < count; i++) {
        printf("Item Code: %d\n",store[i].itemCode);
        printf("Item Name: %s\n", store[i].itemName);
        printf("Expiry Date: %s\n", store[i].expiryDate);
        printf("Quantity Available: %d\n",store[i].quantity);
        printf("Price per Piece: %.2f\n",store[i].price);
        printf("Manufactured Date: %s\n", store[i].manufacturedDate);
        printf("\n");
    }
}

int main() {
    struct Item store[100]; // Assuming there are at most 100 items in the store
    int numItems = 0;
    int choice;
```

```
do {
    printf("\nStore Management System\n");
    printf("1. Add Item\n");
    printf("2. Search Item\n");
    printf("3. Remove Quantity\n");
    printf("4. Remove expired item\n");
    printf("5.Update the item\n");
    printf("6.view item\n");
    printf("7. exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch (choice) {
        case 1: // Add Item
            if (numItems < 100) {
                struct Item newItem;

                printf("Enter item code: ");
                scanf("%d", &newItem.itemCode);
                printf("Enter item name: ");
                scanf("%s", newItem.itemName);
                printf("Enter quantity: ");
                scanf("%d", &newItem.quantity);
                printf("Enter price per piece: ");
                scanf("%f", &newItem.price);
                printf("Enter manufactured date: ");
                scanf("%s", newItem.manufacturedDate);
                printf("Enter expiry date: ");
                scanf("%s", newItem.expiryDate);

                store[numItems] = newItem;
                numItems++;
                printf("Item added successfully.\n");
            } else {
                printf("Store is full. Cannot add more items.\n");
            }
            break;

        case 2: // Search Item
            {
                int searchChoice;
                printf("Search by:\n1. Item Code\n2. Item Name\nEnter choice: ");
                scanf("%d", &searchChoice);

                if (searchChoice == 1) {
                    int itemCode;
                    printf("Enter item code: ");
                    scanf("%d", &itemCode);
```

```
            int index = searchByCode(store, numItems, itemCode);
            if (index != -1) {
                displayItem(store[index]);
            } else {
                printf("Item not found.\n");
            }
        } else if (searchChoice == 2) {
            char itemName[50];
            printf("Enter item name: ");
            scanf("%s", itemName);

            int index = searchByName(store, numItems, itemName);
            if (index != -1) {
                displayItem(store[index]);
            } else {
                printf("Item not found.\n");
            }
        }/* else {
            printf("Invalid choice.\n");
        }*/
    }
    break;

case 3: // Remove Quantity
    {
        int itemCode, quantityToRemove;
        printf("Enter item code: ");
        scanf("%d", &itemCode);

        int index = searchByCode(store, numItems, itemCode);
        if (index != -1) {
            displayItem(store[index]);

            printf("Enter quantity to remove: ");
            scanf("%d", &quantityToRemove);

            if (quantityToRemove <= store[index].quantity) {
                store[index].quantity -= quantityToRemove;
                printf("%d quantity of %s removed.\n", quantityToRemove, store[index].itemName);
            } else {
                printf("Insufficient quantity available.\n");
            }
        } else {
            printf("Item not found.\n");
        }
    }
    break;
```

```
            case 4: // Remove Expired Items
                {
                    char currentDate[20];
                    printf("Enter current date (DD/MM/YYYY): ");
                    scanf("%s", currentDate);
                    removeExpiredItems(store, &numItems, currentDate);
                }
                break;
            case 5:
                int updateCode;
                printf("Enter item code to update: ");
                scanf("%d", &updateCode);
                updateItem(store,numItems, updateCode);
                break;
            case 6:
                viewItems(store,numItems);
                break;
            case 7: // Exit
                printf("Exiting...\n");
                break;

            default:
                printf("Invalid choice.\n");
        }
    } while (choice != 4);

    return 0;
}


//OUTPUT
Store Management System
1. Add Item
2. Search Item
3. Remove Quantity
4. Remove expired item
5.Update the item
6.view item
7. exit
Enter your choice: 1
Enter item code: 1
Enter item name: c
Enter quantity: 10
Enter price per piece: 10
Enter manufactured date: 02/04/2023
Enter expiry date: 10/05/2023
Item added successfully.
```

Store Management System
1. Add Item
2. Search Item
3. Remove Quantity
4. Remove expired item
5.Update the item
6.view item
7. exit
Enter your choice: 4
Enter current date (DD/MM/YYYY): 10/04/2023
No expired items found.