

## MEDICAL EQUIPMENT MANAGEMENT SYSTEM

### 1. PROJECT TITLE:

Medical Equipment Management System.

### 2. PROJECT DESCRIPTION:

The Medical Equipment Management System is a console-based C program designed to manage medical equipment inventory efficiently. It allows users to add new equipment, display all equipment details, search for equipment by ID, update the quantity of existing equipment, and delete equipment records. This system helps hospitals and medical facilities maintain an organized record of their equipment, ensuring availability and proper tracking.

### 3. RESEARCH & BACKGROUND STUDY:

Managing medical equipment manually can lead to errors, misplaced items, and inefficient tracking. With the increasing number of medical devices in healthcare facilities, a computerized system is essential to maintain accurate records. The project is based on fundamental data structures in C, such as arrays and structures, to store and manipulate equipment data. The system uses basic file handling concepts and user input validation to ensure data integrity. This project builds on knowledge of C programming, including functions, loops, conditionals, and string handling.

### 4. SYSTEM DESIGN & IMPLEMENTATION;

#### 4.1 System Required:

- Input new detail of equipment to add.
- Save the input equipment.
- Display all equipment.
- Search for the equipment by equipment id.
- Update or delete the equipment with the equipment id.

#### 4.2 Algorithm:

1. Start program.
2. Show **Menu**:
  - Add Equipment

## MEDICAL EQUIPMENT MANAGEMENT SYSTEM

- Display All Equipments
  - Search Equipment by ID
  - Update Equipment Quantity
  - Delete Equipment
  - Exit
3. User enters choice.
  4. Switch based on choice:
    - If **1** → Call `addEquipment()`
    - If **2** → Call `displayEquipments()`
    - If **3** → Call `searchEquipment()`
    - If **4** → Call `updateQuantity()`
    - If **5** → Call `deleteEquipment()`
    - If **6** → Exit
    - Else → Invalid choice
  5. Repeat until choice = 6.
  6. Stop program.

### 4.3 Program:

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

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

```
#define MAX_EQUIPMENTS 100
```

```
typedef struct {
```

```
    int id;
```

```
    char name[50];
```

```
    int quantity;
```

```
    char department[50];
```

```
} Equipment;
```

```
Equipment equipments[MAX_EQUIPMENTS];
```

```
int count = 0;
```

## MEDICAL EQUIPMENT MANAGEMENT SYSTEM

```
void addEquipment() {
    if (count == MAX_EQUIPMENTS) {
        printf("Storage full! Cannot add more equipment.\n");
        return;
    }
    printf("Enter Equipment ID: ");
    scanf("%d", &equipments[count].id);
    getchar(); // consume newline
    printf("Enter Equipment Name: ");
    fgets(equipments[count].name, sizeof(equipments[count].name), stdin);
    equipments[count].name[strcspn(equipments[count].name, "\n")] = 0; // remove newline
    printf("Enter Quantity: ");
    scanf("%d", &equipments[count].quantity);
    getchar();
    printf("Enter Department: ");
    fgets(equipments[count].department, sizeof(equipments[count].department), stdin);
    equipments[count].department[strcspn(equipments[count].department, "\n")] = 0;
    count++;
    printf("Equipment added successfully!\n");
}

void displayEquipments() {
    if (count == 0) {
        printf("No equipment to display.\n");
        return;
    }
    printf("\n--- Medical Equipment List ---\n");
```

## MEDICAL EQUIPMENT MANAGEMENT SYSTEM

```
printf("%-5s %-20s %-10s %-20s\n", "ID", "Name", "Quantity", "Department");  
for (int i = 0; i < count; i++) {  
    printf("%-5d %-20s %-10d %-20s\n", equipments[i].id, equipments[i].name,  
equipments[i].quantity, equipments[i].department);  
}  
}
```

```
void searchEquipment() {  
    int id;  
    printf("Enter Equipment ID to search: ");  
    scanf("%d", &id);  
    for (int i = 0; i < count; i++) {  
        if (equipments[i].id == id) {  
            printf("Equipment found:\n");  
            printf("ID: %d\nName: %s\nQuantity: %d\nDepartment: %s\n",  
                equipments[i].id, equipments[i].name, equipments[i].quantity,  
equipments[i].department);  
            return;  
        }  
    }  
    printf("Equipment with ID %d not found.\n", id);  
}
```

```
void updateQuantity() {  
    int id, newQty;  
    printf("Enter Equipment ID to update quantity: ");  
    scanf("%d", &id);  
    for (int i = 0; i < count; i++) {
```

## MEDICAL EQUIPMENT MANAGEMENT SYSTEM

```
    if (equipments[i].id == id) {  
        printf("Current quantity: %d\n", equipments[i].quantity);  
        printf("Enter new quantity: ");  
        scanf("%d", &newQty);  
        equipments[i].quantity = newQty;  
        printf("Quantity updated successfully.\n");  
        return;  
    }  
}  
printf("Equipment with ID %d not found.\n", id);  
}
```

```
void deleteEquipment() {  
    int id, index = -1;  
    printf("Enter Equipment ID to delete: ");  
    scanf("%d", &id);  
    for (int i = 0; i < count; i++) {  
        if (equipments[i].id == id) {  
            index = i;  
            break;  
        }  
    }  
    if (index == -1) {  
        printf("Equipment with ID %d not found.\n", id);  
        return;  
    }  
    for (int i = index; i < count - 1; i++) {
```

## MEDICAL EQUIPMENT MANAGEMENT SYSTEM

```
        equipments[i] = equipments[i + 1];
    }
    count--;
    printf("Equipment deleted successfully.\n");
}

int main() {
    int choice;
    do {
        printf("\n--- Medical Equipment Management System ---\n");
        printf("1. Add Equipment\n");
        printf("2. Display All Equipments\n");
        printf("3. Search Equipment by ID\n");
        printf("4. Update Equipment Quantity\n");
        printf("5. Delete Equipment\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        getchar(); // consume newline

        switch (choice) {
            case 1:
                addEquipment();
                break;
            case 2:
                displayEquipments();
                break;
```

## MEDICAL EQUIPMENT MANAGEMENT SYSTEM

```
    case 3:
        searchEquipment();
        break;
    case 4:
        updateQuantity();
        break;
    case 5:
        deleteEquipment();
        break;
    case 6:
        printf("Exiting program.\n");
        break;
    default:
        printf("Invalid choice! Please try again.\n");
}
} while (choice != 6);

return 0;
}
```

## 5. TESTING AND RESULTS:

### 5.1 OUTPUT:

```
--- Medical Equipment Management
System ---
1. Add Equipment
2. Display All Equipments
3. Search Equipment by ID
4. Update Equipment Quantity
5. Delete Equipment
6. Exit
Enter your choice: 1
Enter Equipment ID: 1234
Enter Equipment Name: Injection
Enter Quantity: 5
Enter Department: Medical
Equipment added successfully!
```

```
--- Medical Equipment Management
System ---
1. Add Equipment
2. Display All Equipments
3. Search Equipment by ID
4. Update Equipment Quantity
5. Delete Equipment
6. Exit
Enter your choice: 2

--- Medical Equipment List ---
ID      Name      Quantity
Department
1234 Injection      5
Medical
```

## MEDICAL EQUIPMENT MANAGEMENT SYSTEM

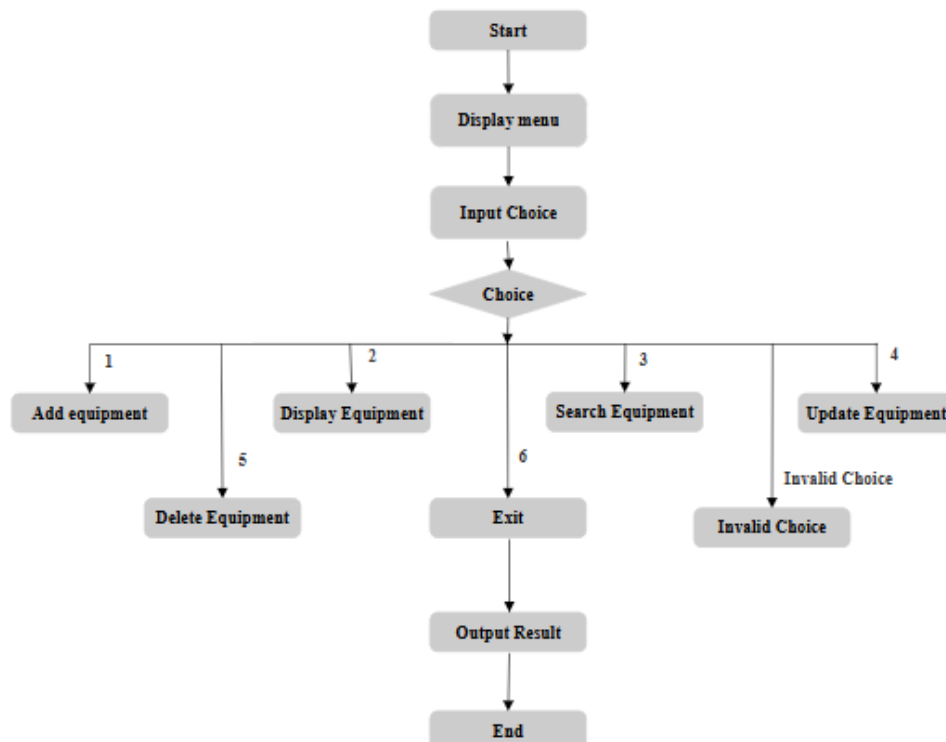
```
--- Medical Equipment Management
System ---
1. Add Equipment
2. Display All Equipments
3. Search Equipment by ID
4. Update Equipment Quantity
5. Delete Equipment
6. Exit
Enter your choice: 3
Enter Equipment ID to search: 1234
Equipment found:
ID: 1234
Name: Injection
Quantity: 5
Department: Medical
```

```
--- Medical Equipment Management
System ---
1. Add Equipment
2. Display All Equipments
3. Search Equipment by ID
4. Update Equipment Quantity
5. Delete Equipment
6. Exit
Enter your choice: 4
Enter Equipment ID to update
quantity: 1234
Current quantity: 5
Enter new quantity: 6
Quantity updated successfully.
```

```
--- Medical Equipment Management
System ---
1. Add Equipment
2. Display All Equipments
3. Search Equipment by ID
4. Update Equipment Quantity
5. Delete Equipment
6. Exit
Enter your choice: 5
Enter Equipment ID to delete: 1234
Equipment deleted successfully.
```

```
--- Medical Equipment Management
System ---
1. Add Equipment
2. Display All Equipments
3. Search Equipment by ID
4. Update Equipment Quantity
5. Delete Equipment
6. Exit
Enter your choice: 6
Exiting program.
```

### 5.2 FLOWCHART:





## MEDICAL EQUIPMENT MANAGEMENT SYSTEM

### 5.3 RESULT:

Test Case No.	Operation	Input	Expected Output	Actual Output	Status
1	Add Equipment	ID = 101, Name = ECG Machine, Qty = 5, Dept = Cardiology	Equipment added successfully	Equipment added successfully	Pass
2	Display Equipments	Choice = 2	Show list with one entry: <b>101 ECG Machine 5 Cardiology</b>	Displayed as expected	Pass
3	Search Equipment	ID = 101	Equipment found with details (ID, Name, Quantity, Department)	Correct details shown	Pass
4	Search (Invalid ID)	ID = 999	Message: "Equipment with ID 999 not found."	Message displayed correctly	Pass
5	Update Quantity	ID = 101, New Qty = 8	Quantity updated successfully	Quantity updated successfully	Pass
6	Delete Equipment	ID = 101	Equipment deleted successfully	Equipment deleted successfully	Pass
7	Display Empty List	Choice = 2 (after deleting all items)	Message: "No equipment to display."	Message displayed correctly	Pass
8	Exit Program	Choice = 6	Program terminates with message "Exiting program."	Program exited as expected	Pass

### 6. CONCLUSION AND FUTURE ENHANCEMENT:

The Medical Equipment Management System was successfully implemented in C to manage hospital equipment efficiently. It allows adding, displaying, searching, updating, and deleting records through a simple menu-driven interface. The system works as expected, reduces manual effort, and can be further enhanced with file storage, database integration, and a graphical interface.