

---

# PROJECT REPORT

---

Course Code: CSEG1032

Course Title: Programming in C

Project Title: MEDIX-pharmacy company management system

Student Name: Princy Agrawal

Sap id: 590024460

Semester: 1st

---

## 1. ABSTRACT

MEDIX is a C-based pharmacy management software designed to automate the daily operations of a pharmacy company.

The system manages inventory, billing, raw materials, and production efficiently. This project demonstrates

modular programming, file handling, array, strings and dynamic memory management in C etc .

The software reduces manual errors, speeds up operations, and provides accurate record keeping.

---

## 2. OBJECTIVE

### THE MAIN OBJECTIVES OF THE MEDIX – PHARMACY COMPANY MANAGEMENT SOFTWARE ARE:

-Automate Pharmacy Operations: Reduce manual effort in managing inventory, billing, raw materials, and production.

-Accurate Inventory Management: Keep real-time track of medicines and raw materials to avoid stock-outs or overstocking.

-Efficient Billing System: Generate accurate bills quickly and maintain records of all transactions.

-Production Tracking: Monitor production batches and ensure raw materials are used efficiently.

-Error Reduction: Minimize human errors in calculations, record-keeping, and stock updates.

-Demonstrate Programming Concepts: Apply modular programming, file handling, dynamic memory management, and pointer-based string manipulation in C.

---

## 3. PROBLEM DEFINITION

### MANUAL MANAGEMENT IS VERY TIME-CONSUMING AND ERROR-PRONE IN A PHARMACY. THE CHALLENGES INCLUDE:

-Tracking medicines and stock manually.

-Calculating bills and maintaining records of transactions.

-Managing raw materials and production batches.

-MEDIX performs all these operations by means of a modular C program, the work process is efficient, reliable, and free from errors.

---

## 4. SYSTEM DESIGN AND ALGORITHM

**THE MEDIX SYSTEM IS DIVIDED INTO 4 MAIN MODULES:**

- Inventory
- Billing
- Raw Materials
- Production
- Utils

**THE PROGRAM SHOWS A MAIN MENU AND PERFORMS OPERATIONS BASED ON USER CHOICE.**

**Module Algorithms :**

### 1. Inventory Module

Open inventory file.  
Show list of medicines.  
User can Add / Update / Delete medicine.  
Save changes and return to menu.

### 2. Billing Module

Ask customer details.  
Show medicines and prices.  
User selects medicines and quantity.  
Calculate total and apply discount.  
Save bill to file and show bill.  
Return to menu.

### 3. Raw Materials Module

Show raw materials and quantity.  
User can Add / Update stock.  
Alert if stock is low.  
Save changes and return to menu.

### 4. Production Module

Input production batch details.  
Check raw material availability.  
Deduct used materials from stock.  
Save production details.  
Return to menu.

### 5. Utils

-clearScreen()

run clear.  
Return to program.

-pauseScreen()

Display message: "Press ENTER to continue... ".  
Wait for user to press ENTER. Return to program.

---

## 5. IMPLEMENTATION DETAILS

### Key Language Features Used:

- Modular with header/source separation.
- File-based data storage.
- Console menu-driven interface for inventory, production, and billing.
- Simple array-based data structures for quick access.

\*MEDIX Pharmacy Management Software is a C-based console application that handles inventory, raw materials, production, and billing. Data is stored in simple .txt files.

## -MODULES-

### -Inventory Management:

Manages medicines (medicines.txt).  
Loads data into arrays and updates stock.

```
typedef struct { char name[50]; int qty; double price; } Medicine;
Medicine medicines[100];
void loadMedicines() { /* read from medicines.txt */ }
```

### -Raw Material Management:

Tracks raw materials (rawmaterials.txt).  
Deducts materials during production.

### -Production Module:

Converts raw materials into medicines.  
Updates inventory and reduces raw material stock.

```
void produceMedicine(char *med, int qty) { /* check and deduct raw materials */ }
```

### -Billing Module:

Generates customer bills (bills.txt).  
Calculates total cost.

```
void generateBill(char *cust, char *med, int qty) { /* compute and save */ }
```

### -Utilities:

Screen clearing, pausing, and menu navigation.

```
void clearScreen() { /* system-specific */ }
void pauseScreen() { printf("Press ENTER... "); getchar(); }
```

---

## 6. TESTING & RESULTS:

---

The MEDIX Pharmacy Software was tested for all functions using sample data files.

### 1. Inventory & Raw Materials

Added and updated medicines and raw materials.

Checked that stock values in files (medicines.txt, rawmaterials.txt) updated correctly.

### 2. Production

Produced medicines using raw materials.

Verified that raw materials decreased and medicine stock increased.

### 3. Billing

Generated bills for customers.

Confirmed that bill details were correctly saved in bills.txt.

### 4. Sample Result

-Before Production:

Medicines: Paracetamol 100, Ibuprofen 50

Raw Materials: ParacetamolPowder 200, IbuprofenPowder 100

Action: Produced 10 Paracetamol

-After Production:

Medicines: Paracetamol 110, Ibuprofen 50

Raw Materials: ParacetamolPowder 190, IbuprofenPowder 100

Bill Generated:

Someone bought 10 Paracetamol → Total ₹25.0

-Observation:

Inventory, production, and billing worked correctly.

Console menus were simple and easy to use.

---

## 7. CONCLUSION AND FUTURE WORK

Conclusion:

- MEDIX Pharmacy Software manages medicines, raw materials, production, and billing.
- Data is saved in files, so nothing is lost when the program closes.
- The program is simple and easy to use.
- All parts of the program work correctly.

#### Future Work:

- Use a database instead of text files for better storage.
  - Add expiry dates of medicines.
  - Check user inputs to avoid mistakes.
  - Add alerts for low stock or production needs.
  - Make the system ready for more users or branches in the future.
- 

## 8. REFERENCES

- YouTube Tutorials: Apna College, CodeWithHarry
  - Class Notes / PPTs provided by the College Faculty
  - Official C Programming Documentation and Examples
- 

## 9. APPENDIX

### -Sample Data Files

#### medicines.txt

6501,Paracetamol,20,30.00  
9155,dolo-650,50,24.00  
5340,Seradon,45,30.00  
2513,Brufan,39,12.00  
3628,Preservatives,600,122.00

#### rawmaterials.txt

9728,Excipients,199  
3636,Solvents,500  
9923,preservatives,600  
1176,Coloring\_agents,1000

#### bills.txt

100,9155,45,1080.00  
100,102,23,460.00  
104,105,23,851.00

### -Sample Code Snippets

#### Add medicine

```
void addMedicine() {  
    Medicine m;  
  
    printf("\n--- Add Medicine ---\n");
```

```
printf("Enter Medicine ID: ");
scanf("%d", &m.id);

printf("Enter medicine name: ");
scanf(" %49[^\\n]", m.name);

printf("Enter quantity: ");
scanf("%d", &m.quantity);

printf("Enter price: ");
scanf("%f", &m.price);

saveMedicineToFile(m);
printf("\nMedicine added successfully!\n");
}
```

### -Sample Console Menu

- 
- 1. Manage Inventory
  - 2. Manage Raw Materials
  - 3. Produce Medicine
  - 4. Generate Bill
  - 5. Exit