# Programming fundamental lab  project Report

## Name:

Ali Abbas 23f-3037

Ahmad 23f-30389

## Section:

BS SE-2A

Problem:-

# Pharmacy System Manage

# Introduction:-

In this project, you can add, modify and delete records of customers, suppliers and medicines. You can also search for customer or supplier details and medicines added into stock. Search can be done by medicine id, customer id or supplier name.

File handling has been extensively and effectively used for almost major functions. The whole project is based on file handling as all medical records are stored in file. Overall, understanding this project will provide you valuable information on how to store, edit, search and delete data using file.

Here, you can input many information like medicine Id, rack no., cabinet no., supplier's name, unit cost, sales price, etc while adding a medicine into the store. You can also view information about report and billing.

# Implementation:-

There are over 10 to 15 functions used in this mini project. I have divided those into the parent functions listed below to help you understand the project better.

Note:- In this system first make 4 to 5 files

Files Name

i) Medicinerecord.txt ii) Customerrecord.txt iii) Supplierrecord.txt iv) Reports.txt

v) Bills.txt

1. Welcome and main menu

void welcome(); at this make a progress bar loading page here

void main_menu();

# 2. Bill slip

void bill(); This function handles the billing process, which typically involves calculating the total cost of medicines purchased by the customer.

# 3. About menu

void about(); This function provides information about the medical store management system, such as its purpose, features, and developers Your name or team name mentioned here.

# 4. Medicine menu

void medicine(); This function manages the medicine-related operations, such as adding new medicines to the inventory, updating medicine details, and searching for medicines.
void medi_sale(); This function handles the sale of medicines to customers, including updating the stock quantity after a sale.
void stock(); This function displays the current stock of medicines available in the inventory.
void update_stock(); This function allows the user to update the stock quantity of existing medicines in the inventory.
void medi_entry(); This function enables the user to add new medicines to the inventory by entering their details.
void medi_search(); This function facilitates searching for medicines based on various criteria such as medicine ID, name, or category.
void remainder(); This function provides a reminder or notification about any pending tasks or alerts related to medicine inventory management.

# 5. Supplier menu

void supplier(); This function is responsible for managing the supplier-related operations, such as adding new suppliers, updating supplier information, and searching for suppliers.
void supp_entry(); This function allows the user to enter information about a new supplier, such as the supplier's name, contact details, and other relevant information.
void supp_list(); This function displays a list of all the suppliers along with their details, such as name, contact information, and any other relevant information.

void sup_update(); This function allows the user to update the information of an existing supplier, such as their contact details or any other relevant information.

void search(); This function is a general search function that can be used to search for various entities within the system, such as medicines, customers, or suppliers.

void search_id();This function allows the user to search for a specific entity (e.g., medicine, customer, supplier) based on its ID.

## 6. Customer menu

void customer();This function is responsible for managing the customer-related operations, such as adding new customers, updating customer information, and searching for customers.

void cust_search(); This function allows the user to search for a specific customer based on various criteria, such as name, ID, or contact information.

void search_cid(); This function allows the user to search for a customer based on their ID.

void search_cname(); This function allows the user to search for a customer based on their name.

void cust_entry(); This function allows the user to enter information about a new customer, such as their name, contact details, and any other relevant information.

void cust_list();This function displays a list of all the customers along with their details, such as name, contact information, and any other relevant information.

void cust_update();This function allows the user to update the information of an existing customer, such as their contact details or any other relevant information.

## 7. Report menu

void report_menu(); This function displays a menu of different types of reports that can be generated from the system.

void report();

void sale_rpt(); This function generates a report of sales transactions.

void sale_rpt_daily(); This function generates a daily sales report.

void profit_rpt(); This function generates a profit report based on the sales and expenses.

void pur_rpt(); This function generates a report of purchases made.

void pur_rpt_daily();This function generates a daily purchase report.

# WORKING:

### Pharmacy Management System Report

### Introduction:

The Pharmacy Management System is designed to streamline the operations of a pharmacy by managing the records of customers, suppliers, and medicines. It utilizes file handling to store, edit, search, and delete data, ensuring efficient data management. The system allows for various operations such as adding new medicine records, modifying existing ones, and deleting obsolete entries. It also supports searching for records based on medicine ID, customer ID, or supplier name.

**Implementation Details** The system is implemented using over 10 to 15 functions, categorized into several menus for ease of use. The primary files created for record-keeping include:

- **Medicinerecord.txt**
- **Customerrecord.txt**
- **Supplierrecord.txt**
- **Reports.txt**
- **Bills.txt**

**Functionality Overview**

1. **Welcome and Main Menu**
    - `void welcome();` - Displays a progress bar loading page.
    - `void main_menu();` - Presents the main menu options to the user.
2. **Bill Slip**
    - `void bill();` - Manages the billing process, calculating the total cost of medicines purchased.
3. **About Menu**
    - `void about();` - Provides information about the system, its purpose, features, and developers.
4. **Medicine Menu**
    - `void medicine();` - Manages medicine inventory operations.
    - `void medi_sale();` - Processes medicine sales and updates stock quantity.
    - `void stock();` - Shows current medicine stock levels.
    - `void update_stock();` - Updates stock quantities for existing medicines.
    - `void medi_entry();` - Adds new medicines to the inventory.
    - `void medi_search();` - Searches for medicines using various criteria.
    - `void remainder();` - Sends reminders or alerts for inventory management tasks.
5. **Supplier Menu**
    - `void supplier();` - Manages supplier-related operations.
    - `void supp_entry();` - Adds new supplier records.
    - `void supp_list();` - Displays a list of suppliers with their details.
    - `void sup_update();` - Updates existing supplier information.
    - `void search();` - General search function for various entities.
    - `void search_id();` - Searches for entities based on their ID.
6. **Customer Menu**
    - `void customer();` - Manages customer-related operations.

- `void cust_search();` - Searches for customers using various criteria.
- `void search_cid();` - Searches for customers by ID.
- `void search_cname();` - Searches for customers by name.
- `void cust_entry();` - Adds new customer records.
- `void cust_list();` - Displays a list of customers with their details.
- `void cust_update();` - Updates existing customer information.

7. **Report Menu**
   - `void report_menu();` - Displays report options.
   - `void report();` - Generates various reports.
   - `void sale_rpt();` - Generates sales transaction reports.
   - `void sale_rpt_daily();` - Generates daily sales reports.
   - `void profit_rpt();` - Generates profit reports.
   - `void pur_rpt();` - Generates purchase reports.
   - `void pur_rpt_daily();` - Generates daily purchase reports.

**Conclusion** The Pharmacy Management System is a comprehensive solution for managing pharmacy operations. Its modular design and extensive use of file handling make it a robust and user-friendly system. The implementation of various functions ensures that all aspects of pharmacy management are covered, from inventory to billing and reporting. In the c++ code I implement all functions according to the function requirements. More ever I add the ouput console color .