**AR Pharmacy Management System**



**Session: 2023-27**

**Submitted by:**

**Salman Naseem 2023-CS-78**

**Supervised by:**

**Dr. Awais Hassan**

**Course**

CSC-102 Programming Fundamentals

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

**CONTENTS**

* **Short Description of Project**
* **Users of Applications**
* **Functional Requirements**
* **Wireframes**
* **Data Structures**
* **Function Prototypes**
* **Flow Diagram**
* **Complete Code**
* **Weakness**
* **Future Directions**

**AR Pharmacy Management System**

**Short Description of Project:**

The AR Pharmacy Management System introduces a dynamic approach to pharmaceutical services, catering to both Admin and Customer users. Admins benefit from streamlined inventory management and Customers enjoy a unique feature where they can articulate symptoms, and the system, employing AR technology, recommends over-the-counter medicines or advises consulting a healthcare professional. Both Admin and Customer users benefit from an interactive virtual assistant for real-time support. The AR Shelf Display enables customers to virtually explore pharmacy products with ease, accessing detailed information about medications. Overall, the system revolutionizes traditional pharmacy management, merging technological innovation with healthcare to deliver a more efficient and personalized service.

**Users of Application:**

There are three users of this system:

* Admin
* Customer

**Functional Requirements:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User Story ID | | User Type | Required Function | Result of Action Performed |
| 1 | Admin | | Manage Staff | Add Employee , remove employee, |
| Add Stock | Add Stock (Medicines) |
| View Stock | View all items present in the stock |
| Remove Stock | Remove , Return items (Medicines) |
| Update Stock | Update Stock (Price, Quantity) |
| Search Stock | Check that is required Medicine present in the stock |
| Customer Order | Admin can place the customer orders |
| Generate Coupan | Generate Coupan Code for Discount |
| View Feedbacks | View Feedbacks of customers |
| View Sale | View the whole sale (also previous sale) |
| Change Password | Admin can change Password but he must remember security key which is only for admin |
| View Profile | View Profile (username , password ,role) |
| Logout | Return back to login page |
| Exit | Exit from the program |
|
|
|

|  |  |  |  |
| --- | --- | --- | --- |
| User Story ID | User Type | Required Function | Result of Action Performed |
| 2 | Customer | View stock | Customer can also view medicine stock |
| Buy Medicine | Buy medicines available in stock |
| System Recommended Medicines | System recommend medicines according to symptoms by the user |
| Feedback | Leave a feedback (may suggestioons) |
| Change password | Change password |
| Logout | Logout and return to login page |
| Exit | Exit from the application |
|
|

* **Admin:**

As Admin , I can:

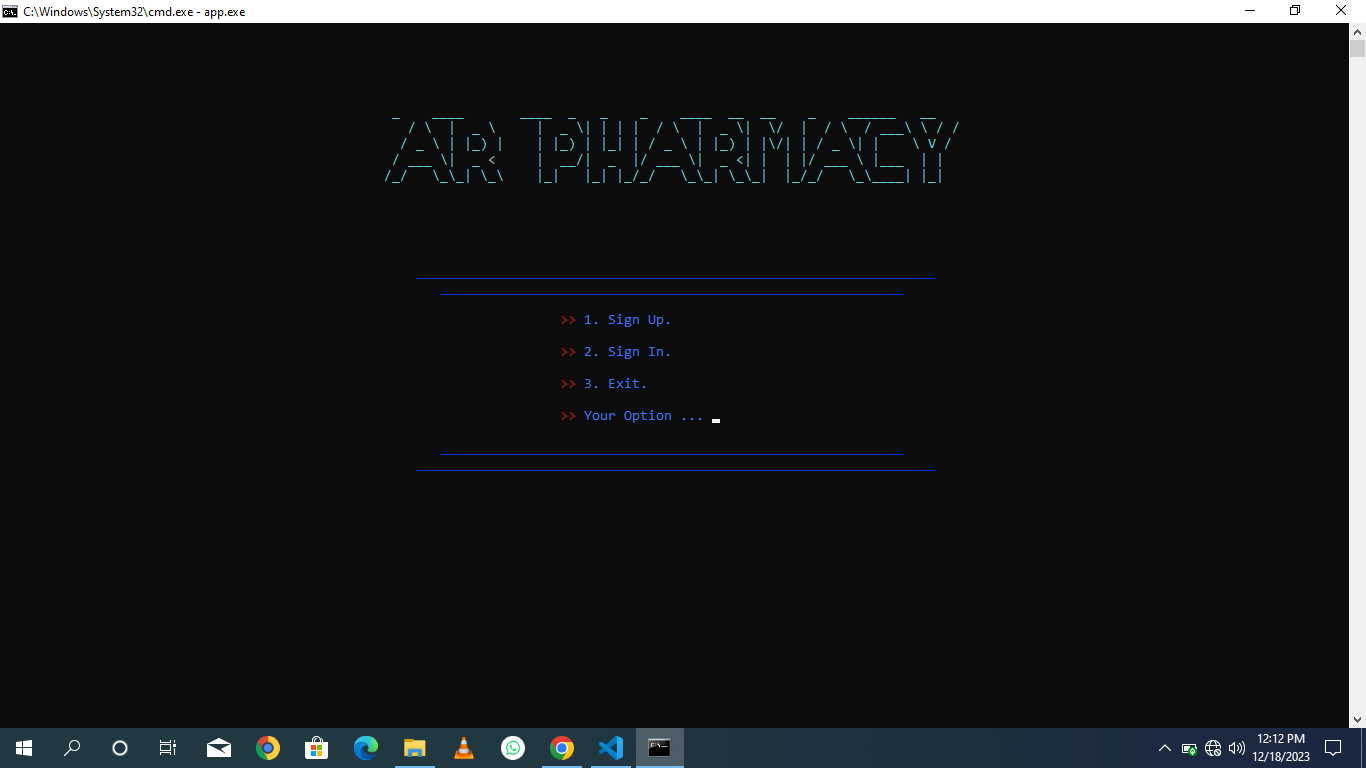
* Manage Staff (Add Employee , Remove Employee , View Employee List)
* View Stock
* Add Stock
* Remove Stock
* Update Stock
* Search Stock (Medicine)
* Place customer order
* Generate Coupan
* View Feedbacks
* View Sale
* Change Password
* View Profile
* **Customer:**

As a customer, I can use self-service channels to:

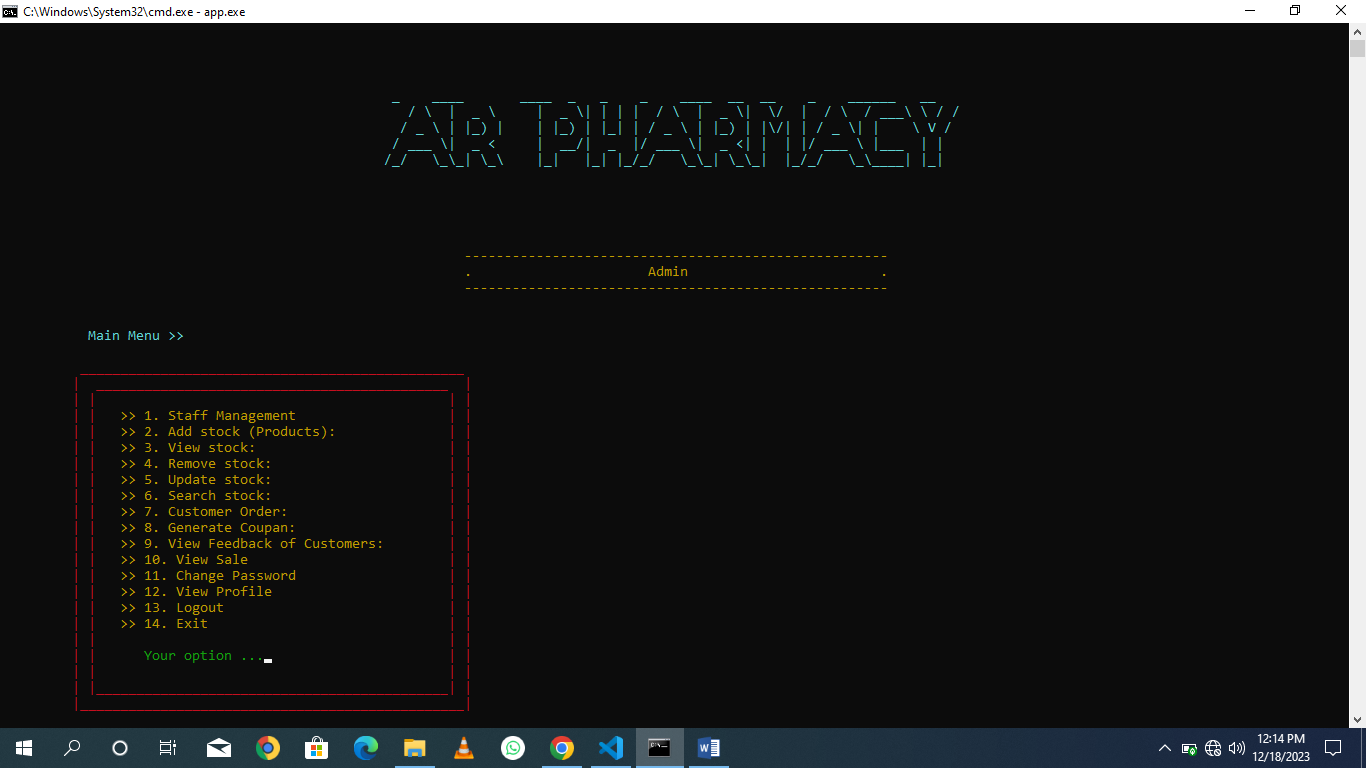
* View Stock (Medicines)
* Buy Medicines
* Avail System Recomeneded Medicines Facility
* Change Password
* View Profile
* Record feedback

**Wireframes:**

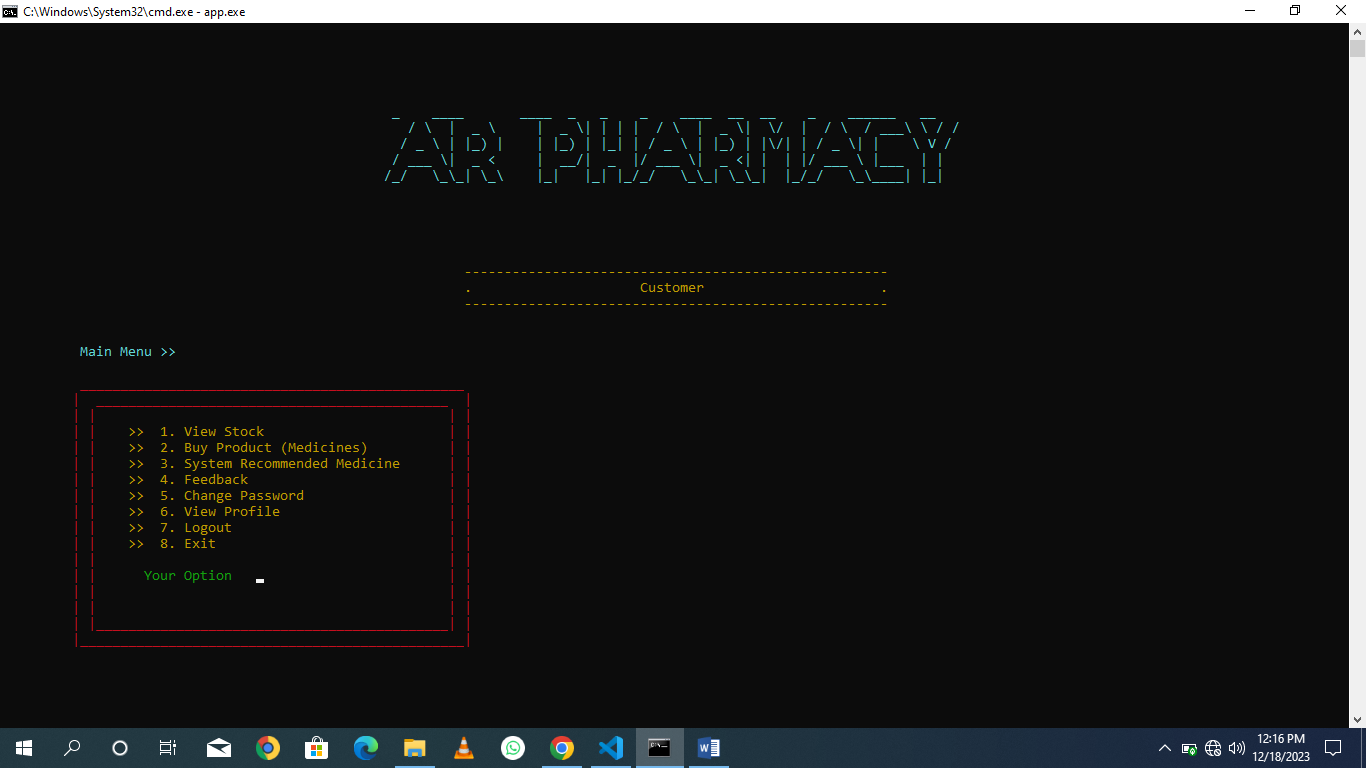
**Main Menu:**

****

**Admin Menu:**

****

**Customer Menu:**

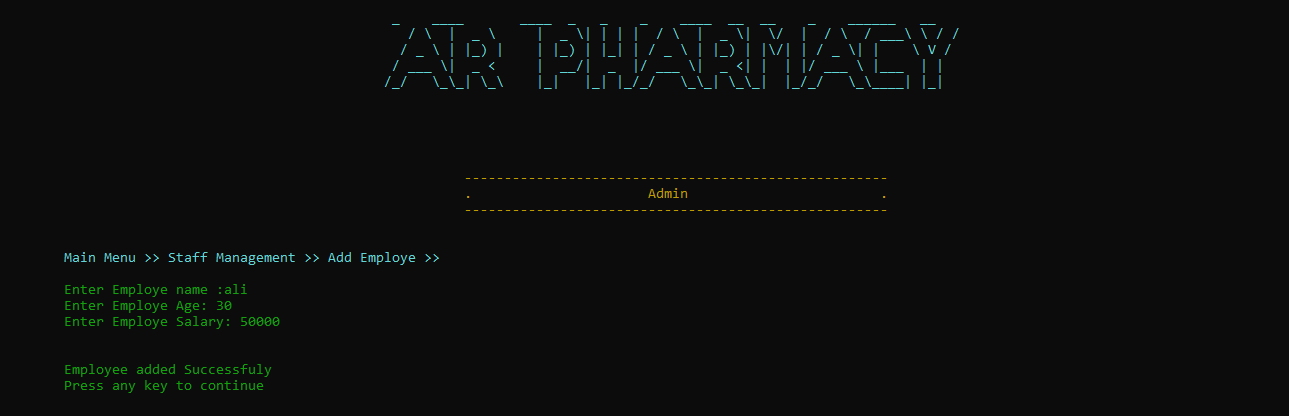
****

**Admin Options:**

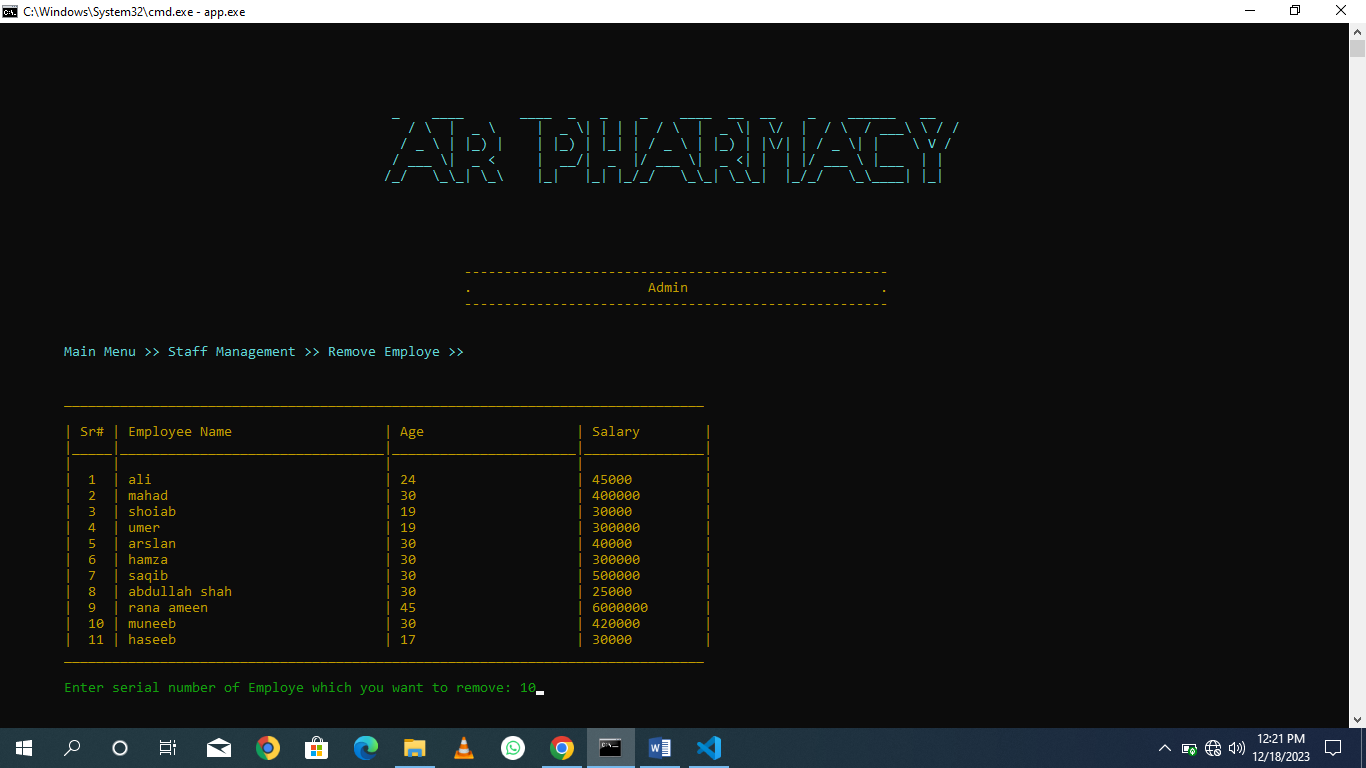
* **Staff Management:**



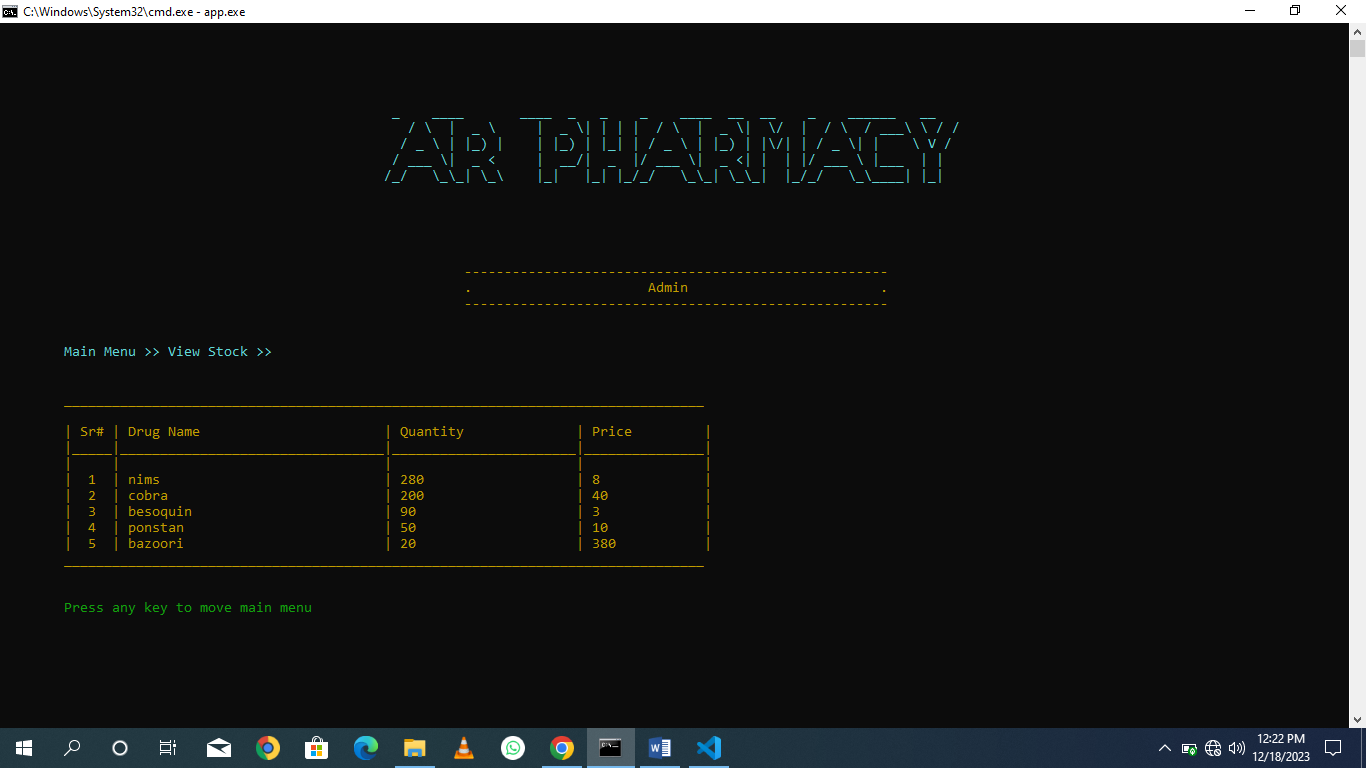
* **Add Employee:**



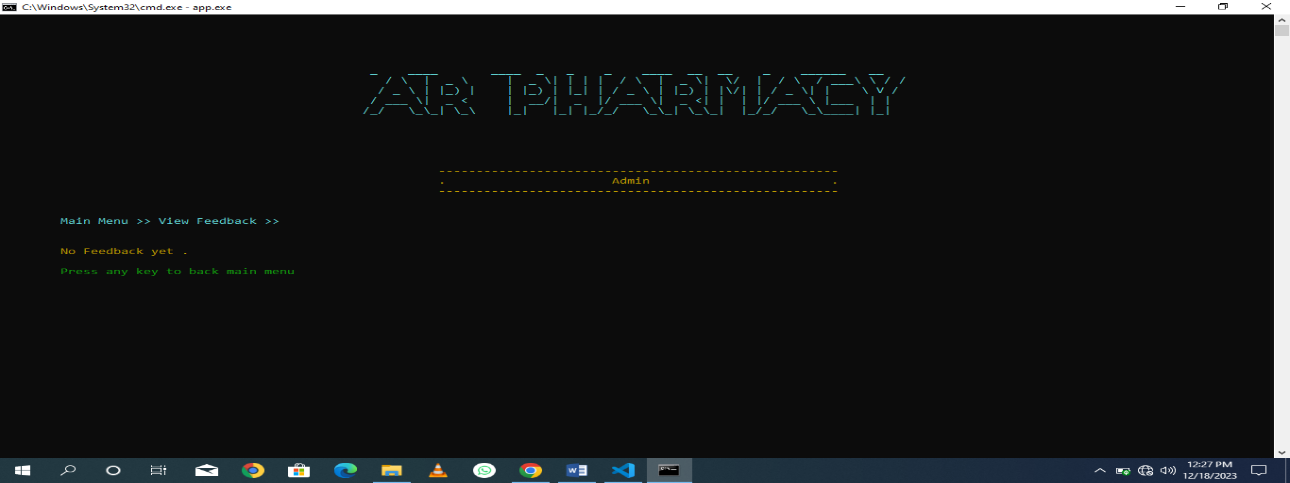
* **Remove Employee:**



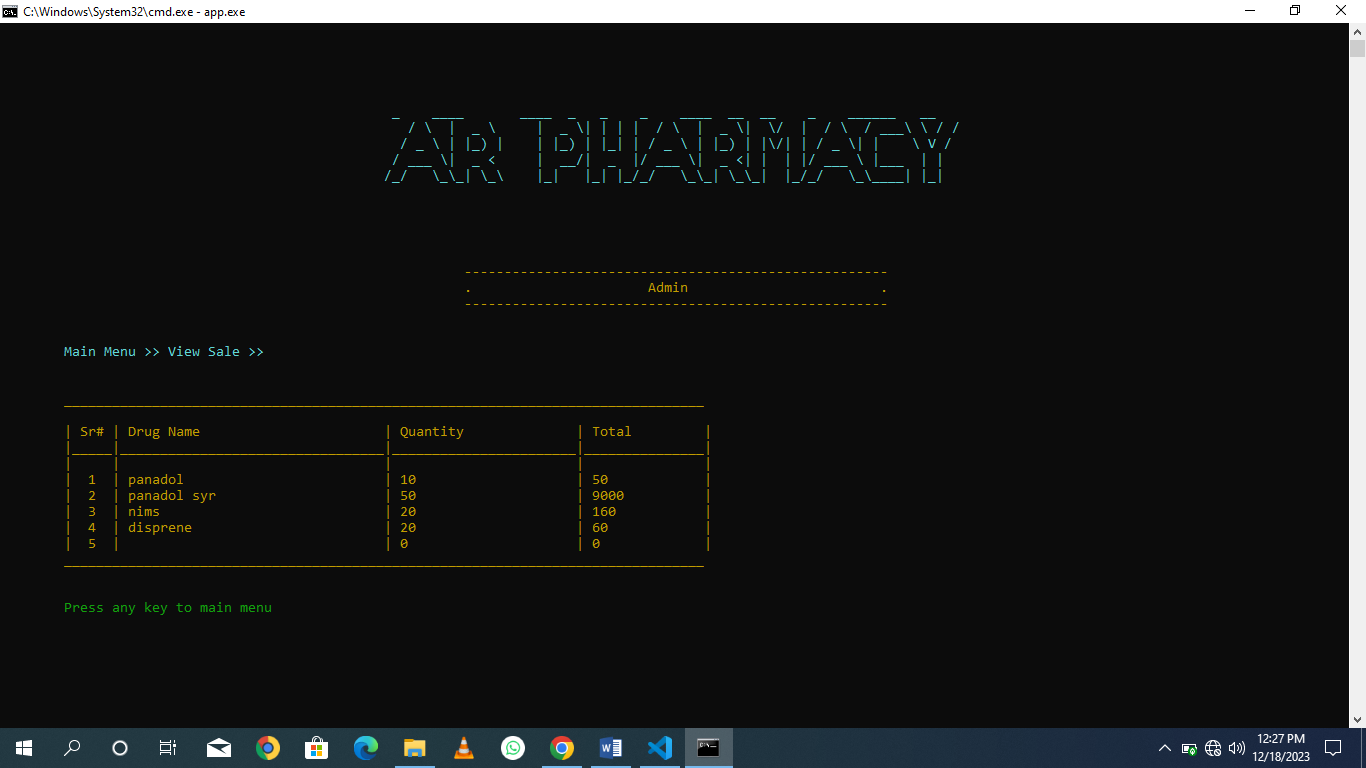
* **View Stock:**

****

* **View Feedbacks of Customer:**

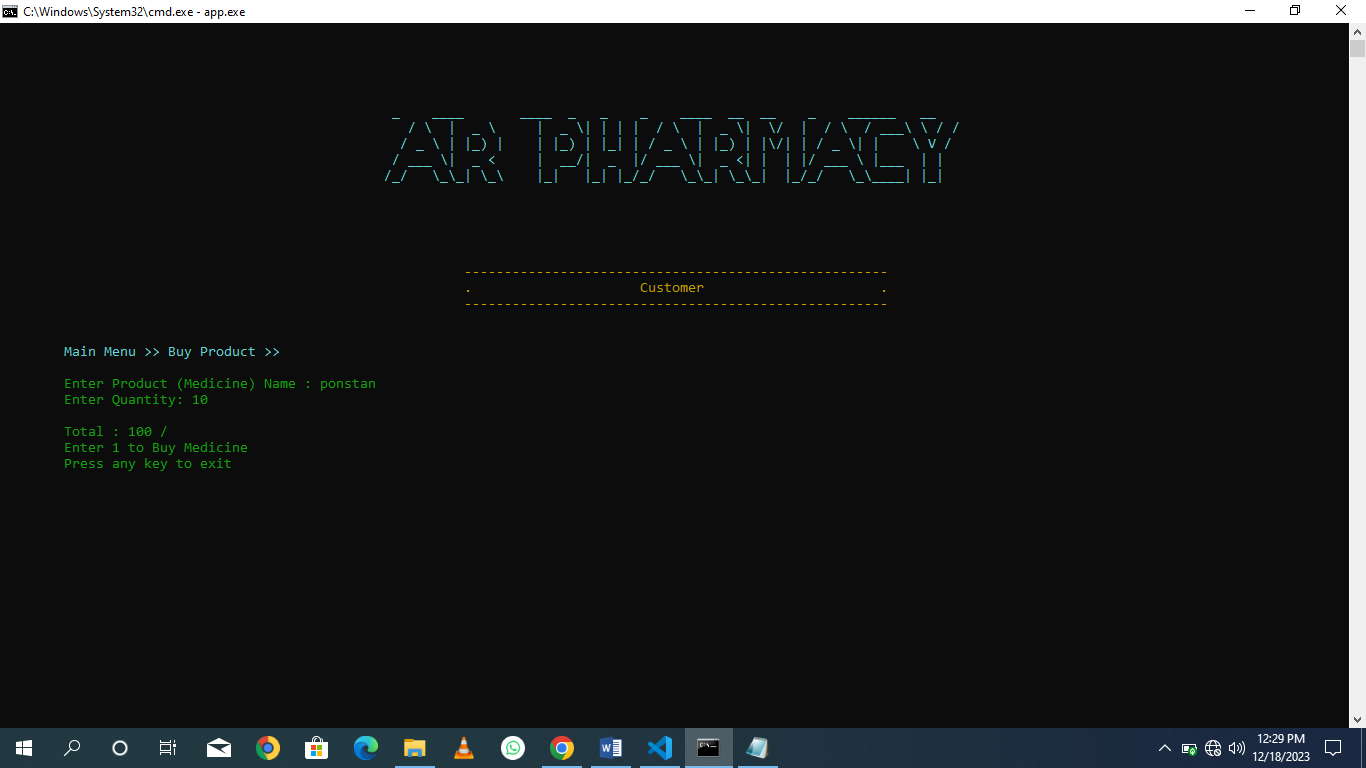
****

* **View Sales:**

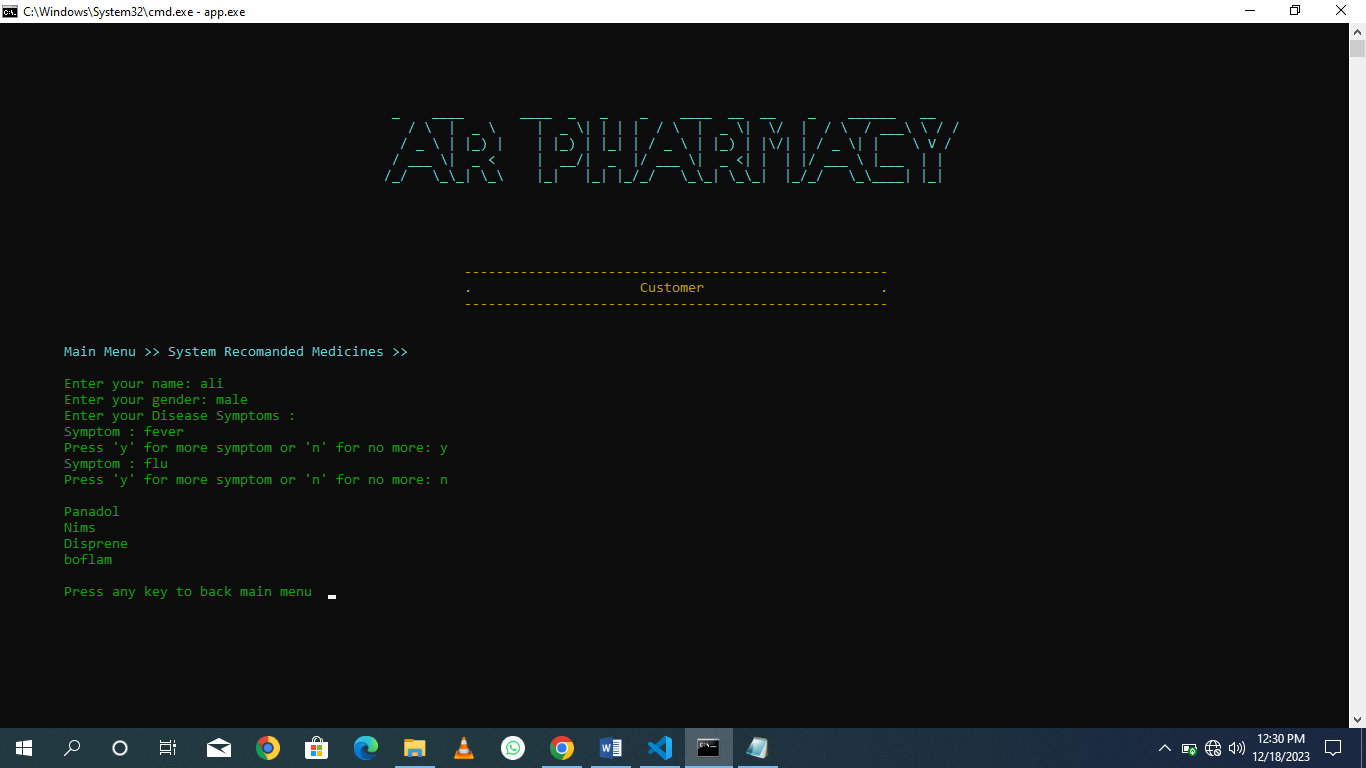
****

**Customer Options:**

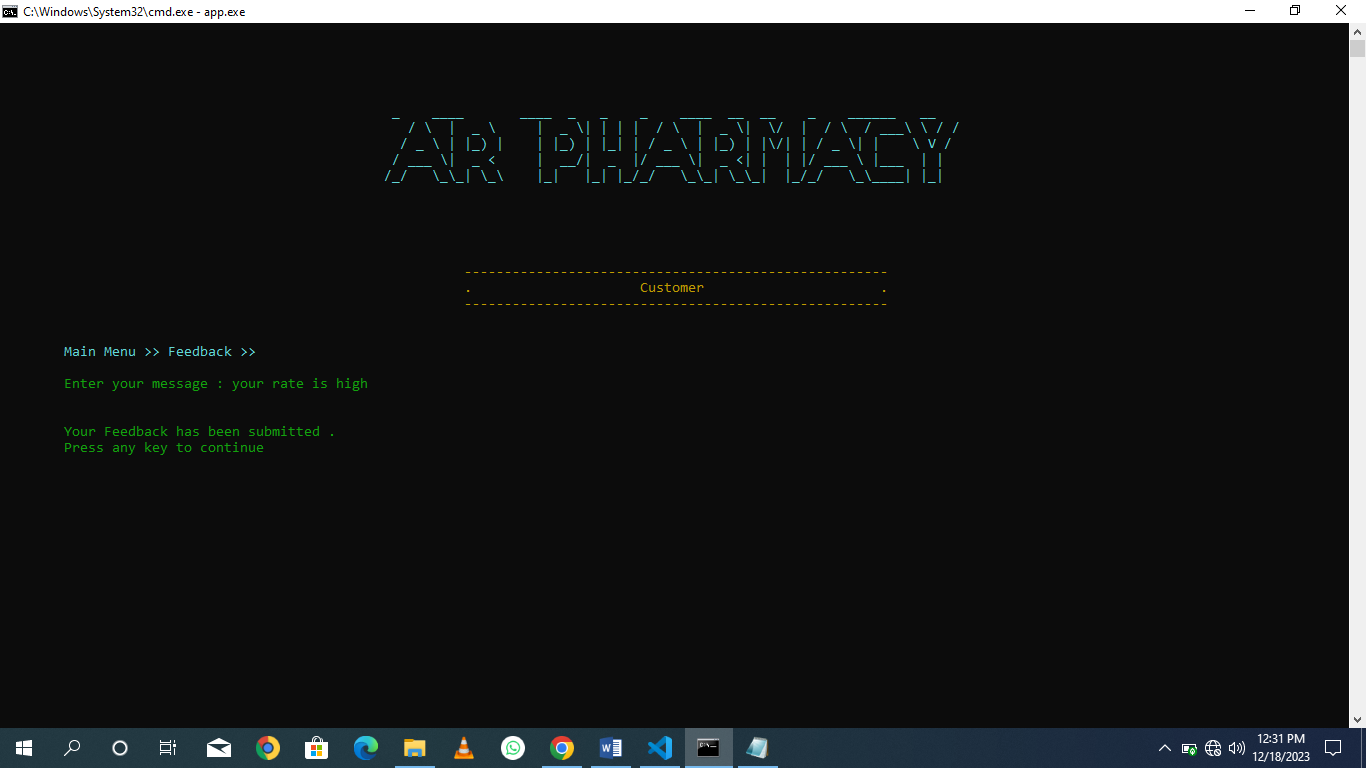
* **Buy Product:**

****

* **System Recommmended Medicines:**

****

* **Record Feedback:**

****

**Data Structures:**

**Arrays:**

* String Name[100]
* String Passoword[100]
* String Role[100]
* String item\_name[100]
* String item\_quantity[100]
* String item\_price[100]
* String empName[100]
* String empAge[100]
* String empSalary[100]
* String sale\_name[100]
* String sale\_quantity[100]
* String sale\_total[100]

**Functions:**

**void headerofApp();**

**void menuHeader();**

**void setColor(int color)**

**{**

**SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), color);**

**}**

**void resetColor()**

**{**

**SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 15);**

**}**

**string menu();**

**void invalidChoice();**

**void signUpHeader();**

**bool signUp(string name, string password, string role, string Name[], string Password[], string Role[]);**

**bool passwordchk(string password);**

**string adminInterface(); // user 1: Admin**

**void adminInterfacebody();**

**string staffInterface(); // submenu of admin user**

**void employelist(string empName[], int empAge[], int empSalary[]);**

**void addEmploye(string empName[], int empAge[], int empSalary[]);**

**bool isInteger(string input);**

**int getIntegerInput(string inputsrt);**

**string stringLength(string str);**

**void removeEmploye(string empName[], int empAge[], int empSalary[], int SrNo);**

**string customerInterface(); // user 2 : customer**

**void customerinterfacebody();**

**void buyProduct(string item\_name[], int item\_quantity[], int item\_price[], string sale\_name[], int sale\_quantity[], int sale\_total[], string coupan, int discount);**

**void systemRecomendation(); // System recommnneded medicines if possible .(unique feature)**

**void recommendMedicines(string name, string gender, string array[], int sympCount);**

**void feedback(string suggestions[], int count); // feedback by user**

**string signIn(string name, string password, string Name[], string Password[], string Role[]);**

**void viewprofile(string Name[], string Password[], string Role[], string profile);**

**void signInAdminheader(); // header show after sign In**

**void signInCustomerheader(); // header show after sign In**

**void enterstock(string item\_name[], int item\_quantity[], int item\_price[]);**

**void removeStock(string item\_name[], int item\_quantity[], int item\_price[], int SrNo);**

**void viewStock(string item\_name[], int item\_quantity[], int item\_price[]);**

**void updateStock(string item\_name[], int item\_quantity[], int item\_price[], int SrNo);**

**void searchStock(string item\_name[], int item\_quantity[], int item\_price[], string search\_name);**

**void generateCoupan(string coupan, int discount);**

**void viewfeedback(string suggestions[], int sug\_count);**

**void viewSale(string sale\_name[], int sale\_quantity[], int sale\_total[]);**

**void logout();**

**void changePassword(string Name[], string Password[], string Role[]);**

**void gotoxy(int x, int y);**

**void load(string Name[], string Password[], string Role[], string empName[], int empAge[], int empSalary[], string item\_name[], int item\_quantity[], int item\_price[], string suggestions[], int sug\_count, string sale\_name[], int sale\_quantity[], int sale\_total[]);**

**string getField(string, int);**

**Flow Chart:**

Main

Admin

Customer

Staff Management

View Stock

View

Buy Products

Add Employee

System Recommended Medicines

Remove Employee

Add Stock

Feedback

Remove Stock

Change Password

View Stock

View Profile

Remove SAtock

Update Stock

Search Stock

Customer Order

Generate Coupan Code

View Feedbacks

View Sale

Change Password

View Profile

**Complete Code:**

#include <iostream>

#include <fstream>

#include <windows.h>

#include <conio.h>

#include <string>

#include <cctype>

using namespace std;

string security\_key = "3120"; //  security key for admin

string option;

fstream file;

int counts = 0;    // for users which have been signed up

int itemCount = 0; // chk how many products have been entered

int empIdx = 0;

int saleIdx = 0;

void headerofApp();

void menuHeader();

void setColor(int color)

{

  SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), color);

}

void resetColor()

{

  SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 15);

}

string menu();

void invalidChoice();

void signUpHeader();

bool signUp(string name, string password, string role, string Name[], string Password[], string Role[]);

bool passwordchk(string password);

string adminInterface(); // user 1: Admin

void adminInterfacebody();

string staffInterface(); // submenu of admin user

void employelist(string empName[], int empAge[], int empSalary[]);

void addEmploye(string empName[], int empAge[], int empSalary[]);

bool isInteger(string input);

int getIntegerInput(string inputsrt);

string stringLength(string str);

void removeEmploye(string empName[], int empAge[], int empSalary[], int SrNo);

string customerInterface(); // user 2 : customer

void customerinterfacebody();

void buyProduct(string item\_name[], int item\_quantity[], int item\_price[], string sale\_name[], int sale\_quantity[], int sale\_total[], string coupan, int discount);

void systemRecomendation(); // System recommnneded medicines if possible .(unique feature)

void recommendMedicines(string name, string gender, string array[], int sympCount);

void feedback(string suggestions[], int count); // feedback by user

string signIn(string name, string password, string Name[], string Password[], string Role[]);

void viewprofile(string Name[], string Password[], string Role[], string profile);

void signInAdminheader();    // header show after sign In

void signInCustomerheader(); // header show after sign In

void enterstock(string item\_name[], int item\_quantity[], int item\_price[]);

void removeStock(string item\_name[], int item\_quantity[], int item\_price[], int SrNo);

void viewStock(string item\_name[], int item\_quantity[], int item\_price[]);

void updateStock(string item\_name[], int item\_quantity[], int item\_price[], int SrNo);

void searchStock(string item\_name[], int item\_quantity[], int item\_price[], string search\_name);

void generateCoupan(string coupan, int discount);

void viewfeedback(string suggestions[], int sug\_count);

void viewSale(string sale\_name[], int sale\_quantity[], int sale\_total[]);

void logout();

void changePassword(string Name[], string Password[], string Role[]);

void gotoxy(int x, int y);

void load(string Name[], string Password[], string Role[], string empName[], int empAge[], int empSalary[], string item\_name[], int item\_quantity[], int item\_price[], string suggestions[], int sug\_count, string sale\_name[], int sale\_quantity[], int sale\_total[]);

string getField(string, int);

main()

{

  string Name[100];

  string Password[100];

  string Role[100];

  string item\_name[100];

  int item\_quantity[100];

  int item\_price[100];

  string empName[100];

  int empAge[100];

  int empSalary[100];

  string sale\_name[100];

  int sale\_quantity[100];

  int sale\_total[100];

  string suggestions[100];

  int SrNo;

  string name, password, role, SI\_name, SI\_password;

  string profile = "", coupan;

  string itemName;

  int itemQuantity, itemPrice;

  int discount;

  int sug\_count = 0; // chk feedback counts

  load(Name, Password, Role, empName, empAge, empSalary, item\_name, item\_quantity, item\_price, suggestions, sug\_count, sale\_name, sale\_quantity, sale\_total);

  while (true)

  {

    system("cls");

    headerofApp();

    menuHeader();

    string menuOption = menu();

    if (menuOption == "1") // sign up

    {

      string s\_key;

      system("cls");

      headerofApp();

      signUpHeader();

      // take input name, password, role;

      setColor(2);

      cout << "\n\tEnter Your Name: ";

      cin.ignore(0);

      getline(cin, name);

      while (true)

      {

        cout << "\tEnter your Password (6 characters): ";

        getline(cin, password);

        bool isPasswordValid = passwordchk(password);

        if (isPasswordValid)

        {

          while (true)

          {

            cout << "\tEnter Your role (A for Admin,C for Customer): ";

            getline(cin, role);

            if (role == "A" || role == "a")

            {

              while (true)

              {

                cout << "\tEnter Admin Security Key: ";

                getline(cin, s\_key);

                if (s\_key == security\_key)

                {

                  break;

                }

                else

                {

                  cout << "\tInvalid Security Key " << endl;

                  Sleep(300);

                }

              }

              break;

            }

            if (role == "C" || role == "c")

            {

              break;

            }

            else

            {

              cout << "\tInvalid Entry ." << endl;

              Sleep(200);

            }

          }

          break;

        }

        else

        {

          cout << "\tInvalid Password ." << endl;

          Sleep(200);

        }

      }

      bool chkSignUp = signUp(name, password, role, Name, Password, Role);

      if (chkSignUp)

      {

        Name[counts] = name;

        Password[counts] = password;

        Role[counts] = role;

        counts++;

        resetColor();

        setColor(4);

        cout << "\n\n\t\tSigned Up successfuly ...." << endl;

        resetColor();

        setColor(2);

        cout << "\t\tPress any key to continue .";

        resetColor();

        file.open("sign up.txt", ios::out);

        for (int i = 0; i < counts; i++)

        {

          file << Name[i] << "," << Password[i] << "," << Role[i] << endl;

        }

        file << endl;

        file.close();

        getch();

      }

      if (!chkSignUp)

      {

        setColor(4);

        cout << "\n\n\t\tThis username already exists ." << endl;

        resetColor();

        setColor(2);

        cout << "\t\tPress any key to continue ...";

        resetColor();

        getch();

      }

    }

    else if (menuOption == "2")

    {

      system("cls");

      headerofApp();

      gotoxy(10, 14);

      setColor(9);

      cout << "\t \_                \_         " << endl;

      cout << "\t| |    \_\_\_   \_\_ \_(\_)\_ \_\_    " << endl;

      cout << "\t| |   / \_ \\ / \_` | | '\_ \\   " << endl;

      cout << "\t| |\_\_| (\_) | (\_| | | | | |  " << endl;

      cout << "\t|\_\_\_\_\_\\\_\_\_/ \\\_\_, |\_|\_| |\_|  " << endl;

      cout << "\t            |\_\_\_/            " << endl;

      resetColor();

      gotoxy(10, 23);

      setColor(2);

      cout << "Enter Your Name: ";

      cin.ignore(0);

      getline(cin, SI\_name);

      profile = SI\_name;

      gotoxy(10, 24);

      cout << "Enter your password: ";

      getline(cin, SI\_password);

      resetColor();

      string chkSignIn = signIn(SI\_name, SI\_password, Name, Password, Role);

      if (chkSignIn == "Admin")

      {

        setColor(4);

        gotoxy(10, 26);

        cout << "Sign In sucsessfuly ...." << endl;

        resetColor();

        Sleep(300);

        while (true)

        {

          string adminOp = adminInterface();

          if (adminOp == "1")

          {

            while (true)

            {

              string staffOp = staffInterface();

              if (staffOp == "1")

              {

                signInAdminheader();

                setColor(11);

                cout << "\n\n\tMain Menu >> Staff Management >> View Employe List >>\n\n";

                resetColor();

                employelist(empName, empAge, empSalary);

                cout << "\n\n\tPress any key to back main menu   ";

                getch();

              }

              else if (staffOp == "2")

              {

                addEmploye(empName, empAge, empSalary);

              }

              else if (staffOp == "3")

              {

                removeEmploye(empName, empAge, empSalary, SrNo);

              }

              else if (staffOp == "4")

              {

                break;

              }

              else if (staffOp == "5")

              {

                return 0;

              }

            }

          }

          else if (adminOp == "2")

          {

            enterstock(item\_name, item\_quantity, item\_price);

          }

          else if (adminOp == "3")

          {

            signInAdminheader();

            setColor(11);

            cout << "\n\n\tMain Menu >> View Stock >>\n\n";

            resetColor();

            viewStock(item\_name, item\_quantity, item\_price);

            cout << "\n\n\tPress any key to move main menu  ";

            getch();

          }

          else if (adminOp == "4")

          {

            removeStock(item\_name, item\_quantity, item\_price, SrNo);

          }

          else if (adminOp == "5")

          {

            updateStock(item\_name, item\_quantity, item\_price, SrNo);

          }

          else if (adminOp == "6")

          {

            string search\_name;

            system("cls");

            headerofApp();

            signInAdminheader();

            setColor(11);

            cout << "\n\n\tMain Menu >> Search Stock (Medicine) >>" << endl;

            resetColor();

            setColor(2);

            cout << "\n\tEnter the name of Medicine :";

            getline(cin, search\_name);

            searchStock(item\_name, item\_quantity, item\_price, search\_name);

            resetColor();

          }

          else if (adminOp == "7")

          {

            buyProduct(item\_name, item\_quantity, item\_price, sale\_name, sale\_quantity, sale\_total, coupan, discount);

          }

          else if (adminOp == "8")

          {

            generateCoupan(coupan, discount);

          }

          else if (adminOp == "9")

          {

            viewfeedback(suggestions, sug\_count);

          }

          else if (adminOp == "10")

          {

            viewSale(sale\_name, sale\_quantity, sale\_total);

          }

          else if (adminOp == "11")

          {

            changePassword(Name, Password, Role);

          }

          else if (adminOp == "12")

          {

            viewprofile(Name, Password, Role, profile);

          }

          else if (adminOp == "13")

          {

            logout();

            break;

          }

          else if (adminOp == "14")

          {

            return 0;

          }

        }

      }

      else if (chkSignIn == "Customer")

      {

        Sleep(200);

        gotoxy(10, 26);

        setColor(4);

        cout << "Sign In sucsessfuly ...." << endl;

        resetColor();

        Sleep(300);

        gotoxy(0, 30);

        while (true)

        {

          string customer\_Op = customerInterface();

          if (customer\_Op == "1")

          {

            signInAdminheader();

            setColor(11);

            cout << "\n\n\tMain Menu >> View Stock >>\n\n";

            resetColor();

            viewStock(item\_name, item\_quantity, item\_price);

            cout << "\n\n\tPress any key to move main menu  ";

            getch();

          }

          else if (customer\_Op == "2")

          {

            buyProduct(item\_name, item\_quantity, item\_price, sale\_name, sale\_quantity, sale\_total, coupan, discount);

          }

          else if (customer\_Op == "3")

          {

            systemRecomendation();

          }

          else if (customer\_Op == "4")

          {

            feedback(suggestions, sug\_count);

          }

          else if (customer\_Op == "5")

          {

            changePassword(Name, Password, Role);

          }

          else if (customer\_Op == "6")

          {

            viewprofile(Name, Password, Role, profile);

          }

          else if (customer\_Op == "7")

          {

            cout<<endl;

            logout();

            break;

          }

          else if (customer\_Op == "8")

          {

            return 0;

          }

        }

      }

      else if (chkSignIn == "invalid")

      {

        setColor(4);

        cout << "\n\t\tYou are not registered .\n\t\tplease sign up first ." << endl;

        resetColor();

        setColor(2);

        cout << "\t\tPress any key to continue ....  ";

        resetColor();

        getch();

      }

      else if (chkSignIn == "password\_incorrect")

      {

        setColor(4);

        gotoxy(10, 27);

        cout << "  Incorrect password" << endl;

        resetColor();

        Sleep(300);

        gotoxy(10, 28);

        setColor(2);

        cout << "Press any key to continue  ";

        resetColor();

        getch();

      }

      else if (chkSignIn == "wrong\_username")

      {

        setColor(4);

        gotoxy(10, 27);

        cout << "  Incorrect Username " << endl;

        resetColor();

        Sleep(300);

        gotoxy(10, 28);

        setColor(2);

        cout << "Press any key to continue  ";

        resetColor();

        getch();

      }

    }

    else if (menuOption == "3")

    {

      return 0;

    }

    else

    {

      gotoxy(73, 25);

      setColor(4);

      cout << "Invalid Choice .";

      resetColor();

      Sleep(300);

    }

  }

}

void gotoxy(int x, int y)

{

  COORD coordinates;

  coordinates.X = x;

  coordinates.Y = y;

  SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coordinates);

}

void invalidChoice()

{

  setColor(4);

  cout << "\tInvalid Choice .";

  Sleep(300);

  resetColor();

}

bool isInteger(string input)

{

  for (int i = 0; input[i] != '\0'; i++)

  {

    if (!isdigit(input[i]))

    {

      return false;

    }

  }

  return true;

}

int getIntegerInput(string inputstr)

{

  string input;

  while (true)

  {

    cout << inputstr;

    getline(cin, input);

    if (input.empty())

    {

      invalidChoice();

      cout << endl;

      setColor(2);

    }

    else if (input.length() >= 8)

    {

      setColor(4);

      cout << "\tInput length exceeded" << endl;

      resetColor();

      setColor(2);

    }

    else if (isInteger(input))

    {

      return stoi(input);

    }

    else

    {

      invalidChoice();

      cout << endl;

      setColor(2);

    }

  }

}

string stringLength(string str)

{

  string input;

  while (true)

  {

    cout << str;

    getline(cin, input);

    if (input.length() <= 20)

    {

      return input;

      break;

    }

    else

    {

      setColor(4);

      cout << "\tInput Length exceeded" << endl;

      resetColor();

      setColor(2);

    }

  }

}

void headerofApp()

{

  setColor(11);

  std::cout << "\n\n\n\n\n";

  cout << "     \t\t\t\t\t\t \_    \_\_\_\_       \_\_\_\_  \_   \_    \_    \_\_\_\_  \_\_  \_\_    \_    \_\_\_\_\_\_   \_\_    " << endl;

  cout << "\t\t\t\t\t\t   / \\  |  \_ \\     |  \_ \\| | | |  / \\  |  \_ \\|  \\/  |  / \\  / \_\_\_\\ \\ / /      " << endl;

  cout << "\t\t\t\t\t\t  / \_ \\ | |\_) |    | |\_) | |\_| | / \_ \\ | |\_) | |\\/| | / \_ \\| |    \\ V /     " << endl;

  cout << "\t\t\t\t\t\t / \_\_\_ \\|  \_ <     |  \_\_/|  \_  |/ \_\_\_ \\|  \_ <| |  | |/ \_\_\_ \\ |\_\_\_  | |      " << endl;

  cout << "\t\t\t\t\t\t/\_/   \\\_\\\_| \\\_\\    |\_|   |\_| |\_/\_/   \\\_\\\_| \\\_\\\_|  |\_/\_/   \\\_\\\_\_\_\_| |\_|      " << endl;

  resetColor();

}

string menu()

{

  string option;

  gotoxy(70, 18);

  cout << "\033[3;31m>>\033[0m \033[1;34m1. Sign Up.\033[0m" << endl;

  gotoxy(70, 20);

  cout << "\033[3;31m>>\033[0m \033[1;34m2. Sign In.\033[0m" << endl;

  gotoxy(70, 22);

  cout << "\033[3;31m>>\033[0m \033[1;34m3. Exit.\033[0m" << endl;

  gotoxy(70, 24);

  cout << "\033[3;31m>>\033[0m \033[1;34mYour Option ... \033[0m";

  setColor(6);

  getline(cin, option);

  resetColor();

  return option;

}

void menuHeader()

{

  int x = 49;

  int y = 14;

  setColor(1);

  gotoxy(x, y = y + 1);

  cout << "   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

  gotoxy(x, y = y + 1);

  cout << "      \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    " << endl;

  gotoxy(x, y = y + 1);

  cout << "                                                                    " << endl;

  gotoxy(x, y = y + 1);

  cout << "                                                                    " << endl;

  gotoxy(x, y = y + 1);

  cout << "                                                                    " << endl;

  gotoxy(x, y = y + 1);

  cout << "                                                                    " << endl;

  gotoxy(x, y = y + 1);

  cout << "                                                                    " << endl;

  gotoxy(x, y = y + 1);

  cout << "                                                                    " << endl;

  gotoxy(x, y = y + 1);

  cout << "                                                                    " << endl;

  gotoxy(x, y = y + 1);

  cout << "                                                                    " << endl;

  gotoxy(x, y = y + 1);

  cout << "                                                                    " << endl;

  gotoxy(x, y = y + 1);

  cout << "      \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    " << endl;

  gotoxy(x, y = y + 1);

  cout << "   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

  resetColor();

}

void signUpHeader()

{

  setColor(6);

  gotoxy(58, 14);

  cout << "-----------------------------------------------------" << endl;

  gotoxy(58, 15);

  cout << ".              Welcome to AR PHARMACY               ." << endl;

  gotoxy(58, 16);

  cout << "-----------------------------------------------------" << endl;

  resetColor();

  cout << endl;

  setColor(9);

  cout << "\t \_\_\_\_  \_                  \_   \_ \_\_\_\_       " << endl;

  cout << "\t/ \_\_\_|(\_) \_\_ \_ \_ \_\_      | | | |  \_ \\      " << endl;

  cout << "\t\\\_\_\_ \\| |/ \_` | '\_ \\     | | | | |\_) |     " << endl;

  cout << "\t \_\_\_) | | (\_| | | | |    | |\_| |  \_\_/      " << endl;

  cout << "\t|\_\_\_\_/|\_|\\\_\_, |\_| |\_|     \\\_\_\_/|\_|         " << endl;

  cout << "\t         |\_\_\_/                             " << endl;

  resetColor();

  cout << endl;

}

bool passwordchk(string password)

{

  if (password.length() == 6)

  {

    return true;

  }

  else

  {

    return false;

  }

}

bool signUp(string name, string password, string role, string Name[], string Password[], string Role[])

{

  for (int i = 0; i <= counts; i++)

  {

    if (name == Name[i])

    {

      return false;

      break;

    }

  }

  return true;

}

void signInAdminheader()

{

  system("cls");

  headerofApp();

  setColor(6);

  gotoxy(58, 15);

  cout << "-----------------------------------------------------" << endl;

  gotoxy(58, 16);

  cout << ".                      Admin                        ." << endl;

  gotoxy(58, 17);

  cout << "-----------------------------------------------------" << endl;

  resetColor();

}

void signInCustomerheader()

{

  system("cls");

  headerofApp();

  setColor(6);

  gotoxy(58, 15);

  cout << "-----------------------------------------------------" << endl;

  gotoxy(58, 16);

  cout << ".                     Customer                      ." << endl;

  gotoxy(58, 17);

  cout << "-----------------------------------------------------" << endl;

  resetColor();

}

string signIn(string SI\_name, string SI\_password, string Name[], string Password[], string Role[])

{

  for (int i = 0; i < counts; i++)

  {

    if (SI\_name == Name[i] && SI\_password == Password[i])

    {

      if (Role[i] == "A" || Role[i] == "a")

      {

        return "Admin";

        break;

      }

      else if (Role[i] == "C" || Role[i] == "c")

      {

        return "Customer";

        break;

      }

    }

    else if (SI\_name == Name[i] && SI\_password != Password[i])

    {

      return "password\_incorrect";

      break;

    }

  }

  for (int i = 0; i < counts; i++)

  {

    if (SI\_password == Password[i] && SI\_name != Name[i])

    {

      return "wrong\_username";

    }

  }

  return "invalid";

}

void viewprofile(string Name[], string Password[], string Role[], string profile)

{

  system("cls");

  headerofApp();

  setColor(11);

  cout << "\n\n\n\tView Profile >> \n"

       << endl;

  resetColor();

  setColor(6);

  for (int i = 0; i < counts; i++)

  {

    if (Name[i] == profile)

    {

      cout << "\n\t\t\t>> Username: " << profile << endl;

      cout << "\t\t\t>> Password: " << Password[i] << endl;

      if (Role[i] == "c")

      {

        cout << "\t\t\t>> Role    : Customer" << endl;

      }

      else

      {

        cout << "\t\t\t>> Role       : Admin" << endl;

      }

    }

  }

  resetColor();

  setColor(2);

  cout << "\n\tPress any key to continue    ";

  getch();

}

void adminInterfacebody(){

int x = 7;

  int y = 21;

setColor(4);

  gotoxy(x, y = y + 1);

  cout << "   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

  gotoxy(x, y = y + 1);

  cout << "  |  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  |" << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| |" << endl;

  gotoxy(x, y = y + 1);

  cout << "  |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

  resetColor();

}

string adminInterface()

{

  while (true)

  {

    signInAdminheader();

    setColor(11);

    cout << "\n\n\t   Main Menu >>\n"

         << endl;

    resetColor();

    adminInterfacebody();

    setColor(6);

    int x=15,y=24;

    gotoxy(x, y = y + 1);

    cout << ">> 1. Staff Management" << endl;

    gotoxy(x, y = y + 1);

    cout << ">> 2. Add stock (Products): " << endl;

    gotoxy(x, y = y + 1);

    cout << ">> 3. View stock: " << endl;

    gotoxy(x, y = y + 1);

    cout << ">> 4. Remove stock: " << endl;

    gotoxy(x, y = y + 1);

    cout << ">> 5. Update stock: " << endl;

    gotoxy(x, y = y + 1);

    cout << ">> 6. Search stock: " << endl;

    gotoxy(x, y = y + 1);

    cout << ">> 7. Customer Order: " << endl;

    gotoxy(x, y = y + 1);

    cout << ">> 8. Generate Coupan: " << endl;

    gotoxy(x, y = y + 1);

    cout << ">> 9. View Feedback of Customers: " << endl;

    gotoxy(x, y = y + 1);

    cout << ">> 10. View Sale  " << endl;

    gotoxy(x, y = y + 1);

    cout << ">> 11. Change Password  " << endl;

    gotoxy(x, y = y + 1);

    cout << ">> 12. View Profile  " << endl;

    gotoxy(x, y = y + 1);

    cout << ">> 13. Logout  " << endl;

    gotoxy(x, y = y + 1);

    cout << ">> 14. Exit  \n" << endl;

    resetColor();

    setColor(2);

    cout << "\t\t  Your option ...";

    getline(cin, option);

    resetColor();

    if (option == "1" || option == "2" || option == "3" || option == "4" || option == "5" || option == "6" || option == "7" || option == "8" || option == "10" || option == "9" || option == "11" || option == "12" || option == "13" || option == "14")

    {

      return option;

      break;

    }

    else

    {

      setColor(4);

      cout<<"\t\t            Invalid Entry";

      Sleep(100);

      resetColor();

      setColor(2);

    }

  }

}

string staffInterface()

{

  while (true)

  {

    signInAdminheader();

    setColor(11);

    cout << "\n\n\t\tMain Menu >> Staff Management >>\n\n"

         << endl;

    resetColor();

    setColor(4);

    cout << "\t\t=======================================\n"

         << endl;

    resetColor();

    setColor(6);

    cout << "\t\t\t1. View Employe list" << endl;

    cout << "\t\t\t2. Add Employe" << endl;

    cout << "\t\t\t3. Remove Employe" << endl;

    cout << "\t\t\t4. Go back to Main Menu " << endl;

    cout << "\t\t\t5. Exit\n"

         << endl;

    resetColor();

    setColor(4);

    cout << "\t\t======================================\n"

         << endl;

    resetColor();

    setColor(6);

    cout << "\t\t\t   Option :  ";

    getline(cin, option);

    resetColor();

    if (option == "1" || option == "2" || option == "3" || option == "4" || option == "5")

    {

      return option;

      break;

    }

    else

    {

      setColor(4);

      cout << "\n\t\t\t    Invalid Choice ";

      resetColor();

      Sleep(300);

    }

  }

}

void employelist(string empName[], int empAge[], int empSalary[])

{

  setColor(6);

  if (empIdx == 0)

  {

    cout << "\n\tNo Employes are in Lists ." << endl;

  }

  if (empIdx != 0)

  {

    cout << "\n\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"

         << endl;

    cout << "\t| Sr# |\tEmployee Name\t\t\t| Age\t\t\t| Salary\t|" << endl;

    cout << "\t|\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

    cout << "\t|     |\t         \t\t\t|         \t\t|      \t\t|" << endl;

    int y = 28;

    for (int i = 0; i < empIdx; i++)

    {

      gotoxy(8, y);

      cout << "|";

      gotoxy(11, y);

      cout << i + 1;

      gotoxy(14, y);

      cout << "|";

      gotoxy(16, y);

      cout << empName[i];

      gotoxy(48, y);

      cout << "|";

      gotoxy(50, y);

      cout << empAge[i];

      gotoxy(72, y);

      cout << "|";

      gotoxy(74, y);

      cout << empSalary[i];

      gotoxy(88, y);

      cout << "|";

      y++;

    }

    cout << "\n\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

  }

  resetColor();

  setColor(2);

}

void addEmploye(string empName[], int empAge[], int empSalary[])

{

  string name;

  signInAdminheader();

  setColor(11);

  cout << "\n\n\tMain Menu >> Staff Management >> Add Employe >> \n\n"

       << endl;

  resetColor();

  setColor(2);

  empName[empIdx] = stringLength("\tEnter Employe name :");

  empAge[empIdx] = getIntegerInput("\tEnter Employe Age: ");

  empSalary[empIdx] = getIntegerInput("\tEnter Employe Salary: ");

  cout << "\n\n\tEmployee added Successfuly " << endl;

  empIdx++;

  file.open("employee data.txt", ios::out);

  for (int i = 0; i < empIdx; i++)

  {

    file << empName[i] << "," << empAge[i] << "," << empSalary[i] << endl;

  }

  file.close();

  cout << "\tPress any key to continue    ";

  getch();

  resetColor();

}

void removeEmploye(string empName[], int empAge[], int empSalary[], int SrNo)

{

  signInAdminheader();

  setColor(11);

  cout << "\n\n\tMain Menu >> Staff Management >> Remove Employe >>\n\n";

  resetColor();

  employelist(empName, empAge, empSalary);cout<<endl;

  setColor(2);

  SrNo = getIntegerInput("\tEnter serial number of Employe which you want to remove: ");

  if (SrNo > 0 && SrNo <= empIdx)

  {

    for (int i = SrNo - 1; i < empIdx; i++)

    {

      empName[i] = empName[i + 1];

      empAge[i] = empAge[i + 1];

      empSalary[i] = empSalary[i + 1];

    }

    empIdx--;

    cout << "\n\n\tEmploye removed successfuly ." << endl;

  }

  else

  {

    invalidChoice();

    cout << endl;

    setColor(2);

  }

  remove("employee data.txt");

  file.open("employee data.txt", ios ::out);

  for (int i = 0; i < empIdx; i++)

  {

    file << empName[i] << "," << empAge[i] << "," << empSalary[i] << endl;

  }

  file.close();

  cout << "\tPress any key to back main menu .";

  getch();

  resetColor();

}

void customerinterfacebody(){

  int x = 7;

  int y = 21;

setColor(4);

gotoxy(x, y = y + 1);

  cout << "   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

  gotoxy(x, y = y + 1);

  cout << "  |  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  |" << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |                                            | |  " << endl;

  gotoxy(x, y = y + 1);

  cout << "  | |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| |" << endl;

  gotoxy(x, y = y + 1);

  cout << "  |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

}

string customerInterface()

{

  while (true)

  {

    signInCustomerheader();

    setColor(11);

    cout << "\n\n\t  Main Menu >>\n"

         << endl;

    resetColor();

    customerinterfacebody();

    setColor(6);

    int x=15,y=24;

    gotoxy(x, y = y + 1);

    cout << "\t>>  1. View Stock " << endl;

    gotoxy(x, y = y + 1);

    cout << "\t>>  2. Buy Product (Medicines)" << endl;

    gotoxy(x, y = y + 1);

    cout << "\t>>  3. System Recommended Medicine" << endl;

    gotoxy(x, y = y + 1);

    cout << "\t>>  4. Feedback" << endl;

    gotoxy(x, y = y + 1);

    cout << "\t>>  5. Change Password " << endl;

    gotoxy(x, y = y + 1);

    cout << "\t>>  6. View Profile " << endl;

    gotoxy(x, y = y + 1);

    cout << "\t>>  7. Logout " << endl;

    gotoxy(x, y = y + 1);

    cout << "\t>>  8. Exit" <<endl;

    resetColor();

   setColor(2);

    cout << "\n\t\t  Your Option   ";

    getline(cin, option);

    resetColor();

    if (option == "1" || option == "2" || option == "3" || option == "4" || option == "5" || option == "6" || option == "7" || option == "8")

    {

      return option;

      break;

    }

    else

    {

      setColor(4);

      cout<<"\n\t\t   Invalid entry";

      resetColor();

      Sleep(150);

    }

  }

}

void buyProduct(string item\_name[], int item\_quantity[], int item\_price[], string sale\_name[], int sale\_quantity[], int sale\_total[], string coupan, int discount)

{

  signInCustomerheader();

  string prod\_name, prod\_quan;

  setColor(11);

  cout << "\n\n\tMain Menu >> Buy Product >>" << endl;

  resetColor();

  setColor(2);

  cout << "\n\tEnter Product (Medicine) Name : ";

  getline(cin, prod\_name);

  bool chk\_name = false;

  for (int i = 0; i < itemCount; i++)

  {

    if (prod\_name == item\_name[i])

    {

      chk\_name = true;

      break;

    }

  }

  if (!chk\_name)

  {

    cout << "\n\n\tSorry this Medicine is not avalaible in Stock ." << endl;

    cout << "\tPress any key to continue .";

    getch();

  }

  if (itemCount != 0)

  {

    for (int i = 0; i < itemCount; i++)

    {

      if (prod\_name == item\_name[i])

      {

        int buy\_quantity;

        buy\_quantity = getIntegerInput("\tEnter Quantity: ");

        if (buy\_quantity <= item\_quantity[i])

        {

          char ch;

          string choice, code;

          int ntotal = buy\_quantity \* item\_price[i];

          cout << "\n\tTotal : " << ntotal << " /" << endl;

          if (ntotal > discount)

          {

            while (true)

            {

              cout << "\tUse Coupan for discount (y/n): ";

              getline(cin, choice);

              if (choice == "y" || choice == "Y")

              {

                cout << "\tEnter Coupan Code: ";

                getline(cin, code);

                if (code == coupan)

                {

                  ntotal -= discount;

                  cout << "\tTotal after discount: " << ntotal << endl;

                  break;

                }

                else

                {

                  cout << "\tInavlid Coupan code " << endl;

                }

              }

              else if (choice == "n" || choice == "N")

              {

                break;

              }

              else

              {

                cout << "\tInvalid choice" << endl;

                Sleep(300);

              }

            }

          }

          cout << "\tEnter 1 to Buy Medicine  " << endl;

          cout << "\tPress any key to exit    ";

          ch = getch();

          if (ch == '1')

          {

            cout << "\n\n\tThanks for buying Medicine .\n\tWish you a nice day " << endl;

            item\_quantity[i] = item\_quantity[i] - buy\_quantity;

            sale\_name[saleIdx] = item\_name[i];

            sale\_quantity[saleIdx] = buy\_quantity;

            sale\_total[saleIdx] = ntotal;

            saleIdx++;

            file.open("sale.txt", ios::out);

            for (int i = 0; i < saleIdx; i++)

            {

              file << sale\_name[i] << "," << sale\_quantity[i] << "," << sale\_total[i] << endl;

            }

            file.close();

            remove("stock data.txt");

            file.open("stock data.txt", ios::out);

            for (int i = 0; i < itemCount; i++)

            {

              file << item\_name[i] << "," << item\_quantity[i] << "," << item\_price[i] << endl;

            }

            file.close();

            cout << "\tPress any key to move main menu   ";

            getch();

            break;

          }

          else

          {

            exit(0);

          }

        }

        else

        {

          cout << "\n\n\tSorry ! Maximum available Stock of this Medicine is " << item\_quantity[i] << " ." << endl;

          cout << "\tPress any key to back mainmenu  ";

          getch();

          break;

        }

        break;

      }

    }

  }

  resetColor();

}

void systemRecomendation()

{

  signInCustomerheader();

  setColor(11);

  cout << "\n\n\tMain Menu >> System Recomanded Medicines >>\n"

       << endl;

  resetColor();

  setColor(2);

  string customerName, customerGender, symptom;

  string symp\_array[5];

  int sympCount = 0;

  cout << "\tEnter your name: ";

  getline(cin, customerName);

  cout << "\tEnter your gender: ";

  getline(cin, customerGender);

  cout << "\tEnter your Disease Symptoms : " << endl;

  string choice;

  while (true)

  {

    cout << "\tSymptom : ";

    getline(cin, symptom);

    symp\_array[sympCount] = symptom;

    sympCount++;

    cout << "\tPress 'y' for more symptom or 'n' for no more: ";

    getline(cin, choice);

    if (choice == "Y" || choice == "y")

    {

      continue;

    }

    else if (choice == "N" || choice == "n")

    {

      break;

    }

    else

    {

      cout << "\tInvalid Choice  " << endl;

    }

  }

  cout << "\n";

  recommendMedicines(customerName, customerGender, symp\_array, sympCount);

  cout << "\n\tPress any key to back main menu  ";

  getch();

  resetColor();

}

void recommendMedicines(string name, string gender, string symp\_array[], int sympCount)

{

  int count = 0;

  for (int i = 0; i < sympCount; i++)

  {

    if (symp\_array[i] == "pain" || symp\_array[i] == "Pain")

    {

      cout << "\tDiclorane" << endl;

      count++;

    }

    if (symp\_array[i] == "fever" || symp\_array[i] == "Fever")

    {

      cout << "\tPanadol\n\tNims" << endl;

      count++;

    }

    if (symp\_array[i] == "flu" || symp\_array[i] == "Flu")

    {

      cout << "\tDisprene\n\tboflam" << endl;

      count++;

    }

    if (symp\_array[i] == "vomting" || symp\_array[i] == "Vomting")

    {

      cout << "\tAntacid" << endl;

      count++;

    }

    if (symp\_array[i] == "headache" || symp\_array[i] == "Headache")

    {

      cout << "\tBesoquin\n\ttonoflex" << endl;

      count++;

    }

    if (symp\_array[i] == "Cough" || symp\_array[i] == "cough")

    {

      cout << "\tCough Syrup" << endl;

      count++;

    }

    if (symp\_array[i] == "bodyache" || symp\_array[i] == "Bodyache")

    {

      cout << "\tIbuprofen\n\tPanadol Extra" << endl;

      count++;

    }

    if (symp\_array[i] == "sorethroat" || symp\_array[i] == "Sorethroat")

    {

      cout << "\tThroat Lozenges\n\tCepacol" << endl;

      count++;

    }

    if (symp\_array[i] == "diarrhea" || symp\_array[i] == "Diarrhea")

    {

      cout << "\tImodium\n\tORS Solution" << endl;

      count++;

    }

    if (symp\_array[i] == "constipation" || symp\_array[i] == "Constipation")

    {

      cout << "\tFiber Supplement\n\tLaxative" << endl;

      count++;

    }

    if (symp\_array[i] == "shortnessofbreath" || symp\_array[i] == "Shortnessofbreath")

    {

      cout << "\tInhaler\n\tBronchodilator" << endl;

      count++;

    }

  }

  if (count == 0)

  {

    cout << "\tSorry, I am unable to recommend you. Please consult a doctor.";

  }

}

void feedback(string suggestions[], int sug\_count)

{

  signInCustomerheader();

  setColor(11);

  cout << "\n\n\tMain Menu >> Feedback >>" << endl;

  resetColor();

  setColor(2);

  cout << "\n\tEnter your message : ";

  getline(cin, suggestions[sug\_count]);

  sug\_count++;

  file.open("feedback.txt", ios::out);

  for (int i = 0; i < sug\_count; i++)

  {

    file << suggestions[i] << endl;

  }

  file.close();

  cout << "\n\n\tYour Feedback has been submitted ." << endl;

  cout << "\tPress any key to continue  ";

  getch();

}

void enterstock(string item\_name[], int item\_quantity[], int item\_price[])

{

  signInAdminheader();

  setColor(11);

  cout << "\n\n\tMain Menu >> Add Stock (Medicines) >>\n\n"

       << endl;

  resetColor();

  setColor(2);

  item\_name[itemCount] = stringLength("\tProduct (Medicine) name: ");

  item\_quantity[itemCount] = getIntegerInput("\tEnter Quantity: ");

  item\_price[itemCount] = getIntegerInput("\tEnter price: ");

  itemCount++;

  file.open("stock data.txt", ios::out);

  for (int i = 0; i < itemCount; i++)

  {

    file << item\_name[i] << "," << item\_quantity[i] << "," << item\_price[i] << endl;

  }

  file.close();

  Sleep(200);

  cout << "\n\n\tProduct added to Stock.\n";

  cout << "\tPress any key to back main menu." << endl;

  getch();

  resetColor();

}

void removeStock(string item\_name[], int item\_quantity[], int item\_price[], int SrNo)

{

  signInAdminheader();

  setColor(11);

  cout << "\n\n\tMain Menu >> Remove Stock >>\n\n";

  resetColor();

  viewStock(item\_name, item\_quantity, item\_price);

  cout << endl;

  setColor(2);

  SrNo = getIntegerInput("\tEnter serial number of product which you want to remove: ");

  if (SrNo > 0 && SrNo <= itemCount)

  {

    for (int i = SrNo - 1; i < itemCount; i++)

    {

      item\_name[i] = item\_name[i + 1];

      item\_quantity[i] = item\_quantity[i + 1];

      item\_price[i] = item\_price[i + 1];

    }

    itemCount--;

    cout << "\n\n\tItem removed ." << endl;

  }

  else

  {

    invalidChoice();cout<<endl;

    setColor(2);

  }

  remove("stock data.txt");

  file.open("stock data.txt", ios::out);

  for (int i = 0; i < itemCount; i++)

  {

    file << item\_name[i] << "," << item\_quantity[i] << "," << item\_price[i] << endl;

  }

  file.close();

  cout << "\tPress any key to back main menu ";

  getch();

  resetColor();

}

void viewStock(string item\_name[], int item\_quantity[], int item\_price[])

{

  if (itemCount == 0)

  {

    setColor(6);

    cout << "\tNo Avalaible Stock ." << endl;

    resetColor();

  }

  else if (itemCount != 0)

  {

    setColor(6);

    cout << "\n\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"

         << endl;

    cout << "\t| Sr# |\tDrug Name\t\t\t| Quantity\t\t| Price\t\t|" << endl;

    cout << "\t|\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

    cout << "\t|     |\t         \t\t\t|         \t\t|      \t\t|" << endl;

    int y = 28;

    for (int i = 0; i < itemCount; i++)

    {

      gotoxy(8, y);

      cout << "|";

      gotoxy(11, y);

      cout << i + 1;

      gotoxy(14, y);

      cout << "|";

      gotoxy(16, y);

      cout << item\_name[i];

      gotoxy(48, y);

      cout << "|";

      gotoxy(50, y);

      cout << item\_quantity[i];

      gotoxy(72, y);

      cout << "|";

      gotoxy(74, y);

      cout << item\_price[i];

      gotoxy(88, y);

      cout << "|";

      y++;

    }

    cout << "\n\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

    resetColor();

  }

  setColor(2);

}

void updateStock(string item\_name[], int item\_quantity[], int item\_price[], int SrNo)

{

  string new\_name;

  int new\_quant, new\_price;

  system("cls");

  headerofApp();

  signInAdminheader();

  setColor(11);

  cout << "\n\n\tMain Menu >> Update Stock >>\n\n";

  resetColor();

  setColor(6);

  if (itemCount == 0)

  {

    cout << "\tNo Avalaible Stock ." << endl;

  }

  else if (itemCount != 0)

  {

    viewStock(item\_name, item\_quantity, item\_price);

    cout << endl;

    setColor(2);

    SrNo = getIntegerInput("\tEnter serial number of product which you want to update: ");

    if (SrNo > 0 && SrNo <= itemCount)

    {

      cout << "\n\tEnter New Name of Drug :";

      getline(cin, new\_name);

      item\_name[SrNo - 1] = new\_name;

      new\_quant = getIntegerInput("\tEnter new Quantity: ");

      item\_quantity[SrNo - 1] = new\_quant;

      new\_price = getIntegerInput("\tEnter new price: ");

      item\_price[SrNo - 1] = new\_price;

      cout << "\n\tStock Updated ." << endl;

      remove("stock data.txt");

      file.open("stock data.txt", ios::out);

      for (int i = 0; i < itemCount; i++)

      {

        file << item\_name[i] << "," << item\_quantity[i] << "," << item\_price[i] << endl;

      }

      file.close();

    }

  else

  {

    invalidChoice();cout<<endl;

    setColor(2);

  }

  }

  cout << "\tPress any key to back main menu   ";

  getch();

  resetColor();

}

void searchStock(string item\_name[], int item\_quantity[], int item\_price[], string search\_name)

{

  setColor(6);

  bool result = false;

  for (int i = 0; i < itemCount; i++)

  {

    if (search\_name == item\_name[i])

    {

      cout << "\n\tYes ! " << item\_name[i] << " is in Stock ." << endl;

      cout << "\tThe quantity is " << item\_quantity[i] << " and Price of one tablet is " << item\_price[i] << " ." << endl;

      result = true;

      break;

    }

  }

  if (result)

  {

    cout << "\n";

  }

  else

  {

    cout << "\n\t" << search\_name << " not found in Stock ." << endl;

  }

  resetColor();

  setColor(2);

  cout << "\n\n\tPress any key to back main menu  ";

  getch();

}

void generateCoupan(string coupan, int discount)

{

  system("cls");

  headerofApp();

  signInAdminheader();

  setColor(11);

  cout << "\n\n\tMain Menu >> Generate Coupan >> \n"

       << endl;

  resetColor();

  setColor(2);

  coupan = stringLength("Generate Coupan code: ");

  discount = getIntegerInput("\tEnter Discount for Coupan: ");

  cout << "\n\tCoupan have generated" << endl;

  cout << "\tPress any key to continue   ";

  cin.ignore();

  getch();

  resetColor();

}

void viewfeedback(string suggestions[], int sug\_count)

{

  signInAdminheader();

  setColor(11);

  cout << "\n\n\tMain Menu >> View Feedback >>\n"

       << endl;

  resetColor();

  setColor(6);

  if (sug\_count == 0)

  {

    cout << "\n\tNo Feedback yet ." << endl;

  }

  for (int i = 0; i < sug\_count; i++)

  {

    cout << "\t" << i + 1 << ". " << suggestions[i] << endl;

  }

  resetColor();

  setColor(2);

  cout << "\n\tPress any key to back main menu  ";

  getch();

  resetColor();

}

void viewSale(string sale\_name[], int sale\_quantity[], int sale\_total[])

{

  signInAdminheader();

  setColor(11);

  cout << "\n\n\tMain Menu >> View Sale >>\n"

       << endl;

  resetColor();

  if (saleIdx == 0)

  {

    setColor(6);

    cout << "\t\t\tNo Sale Record ." << endl;

    resetColor();

  }

  else if (saleIdx != 0)

  {

     setColor(6);

    cout << "\n\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"

         << endl;

    cout << "\t| Sr# |\tDrug Name\t\t\t| Quantity\t\t| Total\t\t|" << endl;

    cout << "\t|\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|" << endl;

    cout << "\t|     |\t         \t\t\t|         \t\t|      \t\t|" << endl;

    int y = 28;

    for (int i = 0; i < itemCount; i++)

    {

      gotoxy(8, y);

      cout << "|";

      gotoxy(11, y);

      cout << i + 1;

      gotoxy(14, y);

      cout << "|";

      gotoxy(16, y);

      cout << sale\_name[i];

      gotoxy(48, y);

      cout << "|";

      gotoxy(50, y);

      cout << sale\_quantity[i];

      gotoxy(72, y);

      cout << "|";

      gotoxy(74, y);

      cout << sale\_total[i];

      gotoxy(88, y);

      cout << "|";

      y++;

    }

    cout << "\n\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

    resetColor();

  }

  setColor(2);

  cout << "\n\n\tPress any key to main menu   ";

  getch();

  resetColor();

}

void changePassword(string Name[], string Password[], string Role[])

{

  system("cls");

  headerofApp();

  signInAdminheader();

  setColor(11);

  cout << "\n\n\tMain Menu >> Change Password >>\n"

       << endl;

  resetColor();

  setColor(2);

  string name, password, new\_password;

  cout << "\tEnter Username: ";

  getline(cin, name);

  cout << "\tEnter your Old Password: ";

  getline(cin, password);

  cout << "\tEnter New Password: ";

  getline(cin, new\_password);

  bool passwordfound = false;

  for (int i = 0; i < counts; i++)

  {

    if (name == Name[i] && password == Password[i])

    {

      Password[i] = new\_password;

      cout << "\n\tPassword changed ." << endl;

      passwordfound = true;

      remove("sign up.txt");

      file.open("sign up.txt", ios::out);

      for (int i = 0; i < counts; i++)

      {

        file << Name[i] << "," << Password[i] << "," << Role[i] << endl;

      }

      file.close();

      break;

    }

  }

  if (!passwordfound)

  {

    cout << "\n\tSomething went wrong " << endl;

  }

  cout << "\tPress any key to continue   ";

  getch();

}

void logout()

{

  setColor(6);

  cout << "\t\t  Loging out";

  for (int i = 0; i < 5; i++)

  {

    cout<<". ";

    Sleep(100);

  }

  resetColor();

}

void load(string Name[], string Password[], string Role[], string empName[], int empAge[], int empSalary[], string item\_name[], int item\_quantity[], int item\_price[], string suggestions[], int sug\_count, string sale\_name[], int sale\_quantity[], int sale\_total[])

{

  string line = "";

  file.open("sign up.txt", ios::in);

  while (!file.eof())

  {

    getline(file, line);

    if (line == "")

    {

      break;

    }

    Name[counts] = getField(line, 0);

    Password[counts] = (getField(line, 1));

    Role[counts] = (getField(line, 2));

    counts++;

  }

  file.close();

  file.open("employee data.txt", ios::in);

  while (!file.eof())

  {

    getline(file, line);

    if (line == "")

    {

      break;

    }

    empName[empIdx] = getField(line, 0);

    empAge[empIdx] = stoi(getField(line, 1));

    empSalary[empIdx] = stoi(getField(line, 2));

    empIdx++;

  }

  file.close();

  file.open("stock data.txt", ios::in);

  while (!file.eof())

  {

    getline(file, line);

    if (line == "")

    {

      break;

    }

    item\_name[itemCount] = getField(line, 0);

    item\_quantity[itemCount] = stoi(getField(line, 1));

    item\_price[itemCount] = stoi(getField(line, 2));

    itemCount++;

  }

  file.close();

  file.open("feedback.txt", ios::in);

  while (!file.eof())

  {

    getline(file, line);

    if (line == "")

    {

      break;

    }

    suggestions[sug\_count] = line;

    sug\_count++;

  }

  file.close();

  file.open("sale.txt", ios::in);

  while (!file.eof())

  {

    getline(file, line);

    if (line == "")

    {

      break;

    }

    sale\_name[saleIdx] = getField(line, 0);

    sale\_quantity[saleIdx] = stoi(getField(line, 1));

    sale\_total[saleIdx] = stoi(getField(line, 2));

    saleIdx++;

  }

  file.close();

}

string getField(string record, int field)

{

  int commaCount = 0;

  string content = "";

  char comma = ',';

  for (int i = 0; i < record.length(); i++)

  {

    if (record[i] == comma)

    {

      commaCount++;

    }

    else if (commaCount == field)

    {

      content += record[i];

    }

  }

  return content;

}

**Weakness:**

* There is limited size of arrays
* High level nesting
* complexity

**Future Directions:**

* Unlimited required array size
* Low nesting as possible as possible
* Reduce complexity of the code
* Make Effective functions