

به نام خدا

گزارش نحوه پیاده سازی پروژه مدیریت رستوران

محمد دهقانی فیروزآبادی

993613028

```

1  #include<iostream>
2
3  #include<cstring>
4
5  #include<conio.h>
6
7  using namespace std;
8
9  struct Database_UP {
10     char User[20];
11     char Pass[20];
12 };
13
14 struct F_PR {
15     char Food_name[30];
16     float Price;
17 };
18
19 void Edit(F_PR[], int &);
20 void Add(F_PR[], int &);
21 void Remove(F_PR[], int &);
22 void UP_File(Database_UP[], int);
23 void Foods_File(F_PR[], int);
24
25

```

قسمت بالای main مختص معرفی کتابخانه ها و تعریف استراکچر ها

و همچنین پروتوتایپ ها برای توابعی که در main استفاده میشود.

تابع Edit مختص تغییر دادن نام و قیمت غذا

تابع Add مختص اضافه کردن یک غذا با قیمتش به منو

تابع Remove مختص حذف یک غذا و قیمتش از منو

تابع UP_File برای ذخیره سازی یوزرنیم و پسورد ها درون فایل

تابع Foods_File برای ذخیره سازی تغییرات اعمال شده بر روی منو درون

فایل

```

int main() {
    FILE * fptr, * ptr;
    int Count_c = 1, Objects_In_Menu = 6, Check_up = 0, i;

    Database_UP Costumers[500];

    //Costumers[0] is specially for admin

    strcpy(Costumers[0].User, "admin");
    /*Default Pass*/
    strcpy(Costumers[0].Pass, "12345678");

    F_PR List[200];
    F_PR Temp;

    //list[20] , Default list of available foods and their prices . Right now 6 objects are available
    //Default list
    strcpy(List[0].Food_name, "Chicken Biryani");
    strcpy(List[1].Food_name, "Chicken Palao");
    strcpy(List[2].Food_name, "Chicken Rice");
    strcpy(List[3].Food_name, "Chicken Burger");
    strcpy(List[4].Food_name, "Nawabi Pizza");
    strcpy(List[5].Food_name, "2.5 Litre Coke");
    List[0].Price = 2;
    List[1].Price = 1.5;
    List[2].Price = 2.5;
    List[3].Price = 1;
    List[4].Price = 4.5;
    List[5].Price = 1.75;

```

در این قسمت دو چیز قابل توجه است

یک اینکه در این پروژه من `Costumers[0].User` و `Costumers[0].Pass` را مختص ادمین قرار دادم و پسورد دیفالت آن 12345678 است که در آینده در صورت درخواست ادمین قابلیت تغییر آن تنها برای یک بار میسر است

و دوم اینکه در اینجا ما منو اولیه که شامل 6 غذا با 6 قیمت است را در اول برنامه میسازیم تا در آینده از آن استفاده کنیم یا که تغییراتی در آن ایجاد کنیم

و اینکه در ابتدا تعداد کاربران این نرم افزار (`Count_c`) 1 نفر در نظر گرفته شده (فقط ادمین) و تعداد غذا های درون منو 6

عدد است

```

fptr = fopen("Food List.txt", "r+");
if (fptr == NULL) {//Save Data When User Start The App For First Time
    fptr = fopen("Food List.txt", "w");
    fprintf(fptr, "%d\n", Objects_In_Menu);
    for (int i = 0; i < Objects_In_Menu; i++) {
        fprintf(fptr, "%s\n", List[i].Food_name);
        fprintf(fptr, "%.2f\n", List[i].Price);
    }
    fclose(fptr);
} else {//Load Data From 'Food List' File
    fscanf(fptr, "%d", & Objects_In_Menu);
    for (int i = 0; i < Objects_In_Menu; i++) {
        fscanf(fptr, "%[^\n]s", List[i].Food_name);
        fscanf(fptr, "%f", & List[i].Price);
    }
    fclose(fptr);
}

ptr = fopen("UP Database.txt", "r+");
if (ptr != NULL) {//Load Data From 'UP Database' ( That Includes All Usernames & Passwords)
    fscanf(ptr, "%d", & Count_c);
    for (int i = 0; i < Count_c; i++) {
        fscanf(ptr, "%[^\n]s", Costumers[i].User);
        fscanf(ptr, "%[^\n]s", Costumers[i].Pass);
    }
}
fclose(ptr);

```

این دو قسمت از دو بخش اساسی این پروژه می باشند

قسمت اول که مربوط به لیست غذا ها است ، مشخص کننده این است که آیا برنامه قبلا یکبار باز شده است یا خیر در صورتی که کاربر برای اولین بار برنامه را اجرا میکند ، لیست منو اولیه که در صفحه قبل توضیح دادیم ، در یک فایل به نام Food List ساخته و ذخیره میشود که خط اول این فایل هم مختص به تعداد غذای موجود در منو است و در صورتی که کاربر قبلا نرم افزار را اجرا کرده باشد و الان برای بار دوم یا بیشتر ، در حال اجرای نرم افزار است ، نرم افزار لیست غذایی که قبلا ایجاد شده بوده است را خوانده و در متغیر های مربوطه ذخیره میکند

قسمت دوم مربوط به دیتابیس یوزر و پسورد ها است

در صورت اجرای برنامه برای اولین بار توسط کاربر ، در اینجا اتفاقی رخ نمیدهد ، اما اگر کاربر در نرم افزار یوزر و پسورد مورد نظر خود را به عنوان یک کاربر جدید قبلا وارد نموده باشد ، فایلی برای او ایجاد شده است (در اسلاید های بعد توضیحات لازم داده شده است) و در اینجا تمام آن یوزر و پسورد ها لود میشوند و توسط برنامه مورد شناسایی قرار میگیرند.

```

switch (First_Menu) {
case 1:
    Signup:
    char Signup;
    system("cls");

    cout << "::[ Restaurant Management System ]::\n\n";
    cout << "Signup As A User Or Admin? (Enter 'u' Or 'a')\n";
    Signup = getch();
    if (Signup == 'u') {
        cout << "\nEnter Your Username: ";
        cin >> Costumers[Count_c].User;

        Check_User:
        for (i = 0; i < Count_c; i++) {
            while (strcmp(Costumers[i].User, Costumers[Count_c].User) == 0) {
                cout << "The Username Already Exists. Please Use A Different Username\n";
                cout << "-----";
                cout << "\nEnter Your Username: ";
                cin >> Costumers[Count_c].User;
                goto Check_User;
            }
        }
        cout << "Enter Your Password: ";
        cin >> Costumers[Count_c].Pass;

        Count_c++;

        cout << "\nCongratulations! You Have Successfully Signed Up";

    } else if (Signup == 'a') {
        if (strcmp(Costumers[0].Pass, "12345678") == 0) {
            cout << "\nYour Username is 'admin'\nPlease Select A Password: ";
            cin >> Costumers[0].Pass;
            cout << "\nCongratulations! You Have Successfully Signed Up";
        } else {
            cout << "You Can't Change Password Of 'admin'! ";
        }
    }

    } else {
        cout << "\nPlease Enter A Valid Digit";
    }

    UP_File(Costumers, Count_c); //Make Database File that includes Username And Passwords.

    cout << "\n\nPress Any Key to Continue...";
    getch();
    system("cls");
    goto Startpage;
}

```

در قسمت بعدی برنامه که مربوط به منو اولیه میباشد (منو شامل Signup و Login) ، من دستورات هر قسمت را در یک کیس قرار دادم

اینجا کیس اول را مشاهده میکنید که در صورت انتخاب گزینه Signup توسط کاربر او وارد قسمت ثبت نام میشود و دو انتخاب دارد . ثبت نام به عنوان کاربر معمولی یا تغییر رمز یوزر ادمین (که تنها یکبار میسر است) .

در صورت ثبت نام موفقیت آمیز تعداد کاربر یکی افزایش می یابد و بلافاصله با تابع UP_File تعداد جدید و یوزرنیم و پسورد جدید درون فایل UP_Database ذخیره میشود .

و پس از اتمام این روند کاربر به صفحه اول برمیگردد.

```

case 2:
    Login_Page:

    cout << "\nEnter Your Username: ";
    cin >> Username;
    cout << "Enter Your Password: ";
    cin >> Password;

    for (i = 0; i <= Count_c; i++) {
        if (strcmp(Costumers[i].User, Username) == 0) { //Searching Username In Our List
            if (strcmp(Costumers[i].Pass, Password) == 0) {
                cout << "\nCongratulations You Have Successfully Logged in";
                cout << "\n\nPress Any Key to Continue...";
                getch();
                system("cls");
            } else {
                cout << "Password Is Incorrect . Please Try Again\n";
                cout << "-----";
                goto Login_Page;
            }
            break;
        }
    }

    if (i == Count_c + 1) {
        cout << "Username Is Not Available . Please Sign Up First ";
        cout << "\n\nPress Any Key to Continue...";
        getch();
        system("cls");
        goto Startpage;
    }
    break;

default:
    cout << "Please Enter A Valid Number";
    cout << "\n\nPress Any Key to Continue...";
    getch();
    system("cls");
    goto Startpage;
}

```

کیس 2 مختص زمانی است که کاربر گزینه Login را وارد میکند . در اینجا ما یوزرنیم و پسورد او را در دو متغیر به نام های Username و Password ذخیره میکنیم .

ابتدا درون استراکچر به دنبال یوزرنیم میگردیم . اگر وجود داشت به مرحله بعد یعنی چک کردن پسورد مختص آن یوزرنیم میرویم . اگر پسورد مطابقت داشت اجازه رفتن به مرحله بعد داده میشود اگر نه دوباره به صفحه ورود برمیگردد.

اگر یوزرنیم درون استراکچر Costumers وجود نداشت به کاربر پیغام میدهیم که اول Signup کنید و او را به صفحه اول برنامه هدایت میکنیم

کیس دیفالت هم برای زمانی است که کاربر به جز گزینه های موجود در منو را انتخاب کند

اینجا نکته مهم این است که در صورتی که یوزرنیم وجود نداشته باشد یا یوزرنیم و پسورد مطابقت نداشته باشد ، اجازه اجرای خط های بعدی کد داده نمیشود

```
//user_mod\
switch (Second_Menu) {

case 1:
for (int i = 0; i < Objects_In_Menu; i++) {
cout << "(" << i + 1 << " " << List[i].Food_name << "----->" << List[i].Price << " only\n";
}
cout << "\n";
break;
case 2:
goto Startpage;
break;
default:
cout << "Please Enter A Valid Number";
cout << "\n\nPress Any Key to Continue...";
getch();
goto Userpage;
}
int Ord_Num;
cout << "Please Select The Order Number: ";
cin >> Ord_Num;
while (Ord_Num > Objects_In_Menu) {
cout << "\nNumber Isn't Valid.Select A Valid Number\n";
cout << "\nPlease Select The Order Number: ";
cin >> Ord_Num;
}
int Num_OF_Deals;
cout << "Please Enter The Number Of Deals: ";
cin >> Num_OF_Deals;

cout << "\n\nOrder: " << List[Ord_Num - 1].Food_name << "\n" << "Number Of Deals: " << Num_OF_Deals << "\nPrice Of Each Deal: $" << List[Ord_Num - 1].Price << " Only\n";
cout << "Total Price: $" << Num_OF_Deals * List[Ord_Num - 1].Price << " Only";
cout << "\n\n-----";
cout << "\n-----Thank You For Coming-----";
cout << "\n\n-----\n\n";
cout << "\n\nPress Any Key to Continue...";
getch();
goto Userpage;
}
```

این مرحله زمانی اجرا میشود که کاربر با موفقیت وارد سیستم شده باشد .

چک میشود که کاربر یا یوزر admin وارد سیستم شده است یا نه . اگر با یوزری به جز یوزر ادمین وارد شده باشد ، قسمت مخصوص به آن اجرا میشود

در صورت ورود کاربر برای او منو مخصوص به خودش را نشان میدهد که شامل دو گزینه See Menu و Logout است

در صورت انتخاب گزینه ای به جز See Menu ، ادامه کد اجرا نمیشود اما اگر گزینه 1 را انتخاب نماید منو غذا نشان داده شده و میتواند سفارش خود را ثبت و در آخر قیمت برای او نشان داده میشود و خرید به پایان میرسد و سپس به صفحه قبل بازمیگردد (صفحه شامل دو گزینه See Menu و Logout)

```

} else { //admin mode
    Admin_Panel: system("cls");
    cout << ":::[ Restaurant Management System ]::\n\n" << "<1> Edit Food Names & Their Prices\n" << "<2> Add A Food To Menu\n" << "<3> Remove A Food From Menu\n" << "<4> Show Menu\n" << "<5> Logout\n\n" << "Please Enter A Number: ";
    cin >> Second_Menu;
    system("cls");
    cout << ":::[ Restaurant Management System ]::\n\n";

    cout << "[-----Editing Panel-----]\n\n\n";
    switch (Second_Menu) {
    case 1:
        Edit(List, Objects_In_Menu);
        cout << "\nThe Changes That You Requested Have Been Applied";
        Foods_File(List, Objects_In_Menu);
        cout << "\n\nPress Any Key To Continue...";
        getch();
        goto Admin_Panel;
        break;
    case 2:
        Add(List, Objects_In_Menu);
        cout << "\nThe Changes That You Requested Have Been Applied";
        Foods_File(List, Objects_In_Menu);
        cout << "\n\nPress Any Key To Continue...";
        getch();
        goto Admin_Panel;
        break;
    case 3:
        Remove(List, Objects_In_Menu);
        cout << "\nThe Changes That You Requested Have Been Applied";
        Foods_File(List, Objects_In_Menu);
        cout << "\n\nPress Any Key To Continue...";
        getch();
        goto Admin_Panel;
    case 4:
        for (int i = 0; i < Objects_In_Menu; i++) {
            cout << "(" << i + 1 << ")" << List[i].Food_name << "-----> $" << List[i].Price << " only\n";
        }
        cout << "\n";
        cout << "\n\nPress Any Key to Go Back To Admin Panel ...";
        getch();
        goto Admin_Panel;
        break;
    case 5:
        goto Startpage;
        break;
    default:
        cout << "\nNumber Isn't Valid.Select A Valid Number\n";
        cout << "\n\nPress Any Key To Continue...";
        getch();
        system("cls");
        goto Admin_Panel;
        break;
    }
}

```

در صورتیکه که کاربر وارد شده ، admin باشد (درون متغیر Username عبارت admin ذخیره شده باشد) این قسمت از کد اجرا میشود که مختص تغییرات منوی غذا توسط ادمین میباشد

در صورت انتخاب هر قسمت تابع مختص به آن اجرا و سپس بلافاصله توسط تابع Foods_File ذخیره سازی مجدد میشود

هرکدام از توابع در اسلاید های بعدی به ترتیب معرفی میشوند

تابع Foods_File

```
void Foods_File(F_PR List[200], int Objects_In_Menu) {  
    FILE * fptr;  
    fptr = fopen("Food List.txt", "w");  
    fprintf(fptr, "%d\n", Objects_In_Menu);  
    for (int i = 0; i < Objects_In_Menu; i++) {  
        fprintf(fptr, "%s\n", List[i].Food_name);  
        fprintf(fptr, "%.2f\n", List[i].Price);  
    }  
    fclose(fptr);  
}
```

این تابع بعد از اعمال تغییرات به روی منوی غذا ها ، بلافاصله فایل Food List رو رونویسی میکند به طوری که ابتدا تعداد غذا های درون لیست که در پروژه با متغیر Objects_In_Menu با آن کار شده ، را در خط اول فایل ذخیره و نام و قیمت غذا را در خط های بعدی در زیر هم قرار میگیرند

تابع Edit

```
void Edit(F_PR List[200], int &Objects_In_Menu) {
    int Option;
    char yes_or_no, Yes = 'Y', yes = 'y', No = 'N', no = 'n', New_Food[30], New_Price;

    for (int i = 0; i < Objects_In_Menu; i++) {
        cout << "(" << i + 1 << ")" << " " << List[i].Food_name << "-----> $" << List[i].Price << "\n";
    }
    cout << "\n\nWhich Item Do You Want To Change?";
    cout << "\nPlease Choose It's Number: ";
    cin >> Option;

    while (Option > Objects_In_Menu) {
        cout << "\nPlease Choose An Available Item: ";
        cin >> Option;
    }

    cout << "\nDo You Want to Change It's Name (Y/N)";
    yes_or_no = getch();

    if (yes_or_no == 'y' || yes_or_no == 'Y') {
        cout << "\nPlease Enter A New Food Name: ";
        scanf("%[^\\n]s", & New_Food);
        strcpy(List[Option - 1].Food_name, New_Food);
    }

    cout << "\nDo You Want to Change It's Price (Y/N)";
    yes_or_no = getch();
    if (yes_or_no == 'y' || yes_or_no == 'Y') {
        cout << "\nPlease Enter A New Price: ";
        cin >> List[Option - 1].Price;
    }
}
```

این تابع با دریافت استراکچر F_PR که مربوط به نام غذا ها و قیمت آنهاست ، ابتدا تمام آنها را نشان میدهد ، سپس با درخواست ادمین و انتخاب یکی از غذا های درون منو ، قابلیت تغییر نام غذا یا تغییر قیمت آن یا هردو را دارد

تابع Add

```
void Add(F_PR List[200], int & Objects_In_Menu) {  
    char New_Food[30], New_Price;  
    cout << "\nPlease Enter A New Food Name: ";  
    scanf("%[^\\n]s", & List[Objects_In_Menu].Food_name);  
    cout << "\nPlease Enter It's Price: ";  
    cin >> List[Objects_In_Menu].Price;  
    Objects_In_Menu++;  
}
```

این تابع همانطور که از اسمش مشخص است با دریافت نام غذا و قیمت آن ، آن را به انتهای منو اضافه میکند و در انتها متغیر Objects_In_Menu که به صورت Call By Reference در اینجا استفاده میشود ، به دلیل افزایش غذای درون منو ، یکی افزایش می یابد

تابع Remove

```
void Remove(F_PR List[200], int & Objects_In_Menu) {
    int Option;

    for (int i = 0; i < Objects_In_Menu; i++) {
        cout << "(" << i + 1 << " ) " << List[i].Food_name << "-----> $" << List[i].Price << "\n";
    }
    cout << "\n\nWhich Item Do You Want To Remove From Menu?";
    cout << "\nPlease Choose It's Number: ";
    cin >> Option;
    for (int i = Option; i < Objects_In_Menu; i++) {
        strcpy(List[i-1].Food_name, List[i].Food_name);
        List[i-1].Price = List[i].Price;
    }

    Objects_In_Menu--;
}
```

در این تابع ابتدا لیست نمایان میشود و سپس ادمین با انتخاب یکی از غذا ها ، آن را به کل از لیست حذف میکند و سپس از لیست غذا ها یکی کم میشود (متغیر Objects_In_Menu)

تابع UP_FILE

```
void UP_File(Database_UP Costumers[500], int Count_c) {  
    FILE * fptr;  
    fptr = fopen("UP Database.txt", "w");  
    fprintf(fptr, "%d\n", Count_c);  
    for (int i = 0; i < Count_c; i++) {  
        fprintf(fptr, "%s\n", Costumers[i].User);  
        fprintf(fptr, "%s\n", Costumers[i].Pass);  
    }  
    fclose(fptr);  
}
```

وظیفه این تابع همانطور که قبلا گفته شد ، ذخیره سازی عناصر موجود در استراکچر Costumers درون فایل مربوط به دیتابیس یوزرنیم و پسورد ها است که بعد از هر Signup رو نویسی میشود

Count_c هم تعداد کاربران موجود در سیستم است که هر بار رونویسی میشود