



Programming in C

Lab Project (Semester-1)

UNIVERSITY MANAGEMENT

Members:

Shivanshu Singh (B220051)

Shripati Pandey (B220052)

Shubhasai Mohapatra (B220053)

International Institute of Information Technology

Bhubaneswar, India

Contents

Title of Project	4
Description of Project	4
Users of the System	4
Facilities provided by the system	4
Code	4
Sample Outputs	26
Contribution by group members	27
References	28

Title of Project

UNIVERSITY MANAGEMENT

Description of Project

This project named University Management is a C based project by the help of which we can manage data of the University Like Students Details, Faculty Details, Staff Details along with a Noticeboard Manager. By the help of this Project we can Add, Delete, Modify, View Details and Search and view Individual details of STUDENT, TEACHER, STAFFS. In This we can also Add view and edit Notice Board also. The Header files includes in the code are:

```
#include<stdio.h> #include<string.h> #include<stdlib.h> #include<conio.h>
```

Users of the System

This project is designed for the University Management so the ADMIN of the university can register and login manage the data for the UNIVERSITY.

Facilities provided by the system

Necessary Features:

- Add student record
- Update records
- Search and view records
- Delete records
- Display a menu for these functions along with exit

Additional Features:

- Notice Board:
- Notice Editing, Viewing and Adding

Code

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<conio.h>
struct logindata
{
    char password[10];
    char username[10];
}ld;
struct student
{
    int id;
    char dept[20];
    char name[20];
    char adress[100];
    float grade;
};
struct faculties
{
    int id;
    char dept[20];
    char name[20];
    char adress[100];
    char number[10];
};
struct staffs
{
    int id;
    char dept[20];
    char name[20];
    char adress[100];
    char number[10];
};
struct notice
{
    int slno;
    char details[200];
};

int mainmenu();
void registration();
int login();
void password();
void students();
int studentmenu();
void staff();
int staffmenu();
void faculty();
```

```

int facultymenu();
//student
void addstudent(FILE * fp);
void modifystudent(FILE * fp);
void displaystudent(FILE * fp);
void Individualstudent(FILE *fp);
FILE * delstudent(FILE * fp);
// faculty
void addfaculty(FILE * fp);
void modifyfaculty(FILE * fp);
void displayfaculty(FILE * fp);
void Individualfaculty(FILE *fp);
FILE * delfaculty(FILE * fp);
//Staff
void addstaff(FILE * fp);
void modifystaff(FILE * fp);
void displaystaff(FILE * fp);
void Individualstaff(FILE *fp);
FILE * delstaff(FILE * fp);
//Notice Board
int noticeboard();
void addnotice(FILE *np);
void viewnotice(FILE *np);
void editnotice(FILE *np);

void loginregistrationmenu();
// void signout();

int main(){
    loginregistrationmenu();
    return 0;
}
void registration(){
    FILE *ptr;
    ptr = fopen("pass.txt","w+");
    printf("Enter Username\n");
    fflush(stdin);
    gets(ld.username);
    printf("Create Password\n");
    fflush(stdin);
    gets(ld.password);
    fprintf(ptr,"%s %s",ld.username,ld.password);
    fclose(ptr);
    printf("Successfully Register\n");
    getch();
    system("cls");
    login();
}

```

```

int login(){
    FILE *ptr;
    char inputusername[10];
    char inputpassword[10];
    char un1[10];
    char pass1[10];
    ptr = fopen("pass.txt","r+");
    printf("Enter User Name:\n");
    fflush(stdin);
    gets(inputusername);
    printf("Enter password:\n");
    scanf("%s",&inputpassword);
    fscanf(ptr,"%s %s",un1,pass1);
    // printf("%s%s",un1,pass1);
    if(strcmp(inputpassword,pass1)==0 && strcmp(inputusername,un1) == 0
){
    int cont = 1;
    printf("Press Enter to Continue");
    getch();
    system("cls");
    while (cont!=0)
    {
        printf("WELCOME\n");
        int choice;
        choice = mainmenu();
        FILE *fps;
        if((fps=fopen("studentdetails.txt","rb+"))==NULL)
        {
            if((fps=fopen("studentdetails.txt","wb+"))==NUL
L)
            {
                printf("Can't create or open Database.");
            }
        }

        FILE *fpt;
        if((fpt=fopen("facultydetails.txt","rb+"))==NULL)
        {
            if((fpt=fopen("facultydetails.txt","wb+"))==NUL
L)
            {
                printf("Can't create or open Database.");
                return 0;
            }
        }
        FILE *fpe;
        if((fpe=fopen("staffdetails.txt","rb+"))==NULL)
        {

```

```

        if((fpe=fopen("staffdetails.txt","wb+"))==NULL)
        {
            printf("Can't create or open Database.");
            return 0;
        }
    }
    FILE *fpn;
    if((fpn=fopen("notice.txt","rb+"))==NULL)
    {
        if((fpn=fopen("notice.txt","wb+"))==NULL)
        {
            printf("Can't create or open Database.");
            return 0;
        }
    }
    switch (choice)
    {
    case 1:
        printf("welcome to Student Section:\n");
        int ch = studentmenu();
        if(ch==1){
            addstudent(fps);
        }
        else if(ch==2){
            displaystudent(fps);
        }
        else if(ch==3){
            Individualstudent(fps);
        }
        else if(ch==4){
            modifystudent(fps);
        }
        else if(ch==5){
            fps=delstudent(fps);
        }
        else
        {
            printf("Wrong Input\n");
        }

        break;
    case 2:
        printf("welcome to Teachers Section:\n");
        int cht = facultymenu();
        if(cht==1){
            addfaculty(fpt);
        }
        else if(cht==2){

```



```

        displayfaculty(fpt);
    }
    else if(cht==3){
        Individualfaculty(fpt);
    }
    else if(cht==4){
        modifyfaculty(fpt);
    }
    else if(cht==5){
        fpt=delfaculty(fpt);
    }
    else
    {
        printf("Wrong Input\n");
    }
    break;
case 3:
    printf("Welcome to Staffs Section:\n");
    int che = staffmenu();
    if(che==1){
        addstaff(fpe);
    }
    else if(che==2){
        displaystaff(fpe);
    }
    else if(che==3){
        Individualstaff(fpe);
    }
    else if(che==4){
        modifystaff(fpe);
    }
    else if(che==5){
        fpe=delstaff(fpe);
    }
    else
    {
        printf("Wrong Input\n");
    }
    break;
case 4:
    printf("Welcome to NoticeBoard Section:\n");
    int chn = noticeboard();
    if(chn==1){
        addnotice(fpn);
    }
    else if (chn==2)
    {
        viewnotice(fpn);
    }

```

```

        }
        else if (chn==3)
        {
            editnotice(fpn);
        }
        else
        {
            printf("Wrong Input\n");
        }

        break;
    case 5:
        printf("Signout\n");
        int c;
        printf("Enter 0 to Confirm Exit\n");
        scanf("%d",&c);
        exit(c);
        break;
    default:
        printf("Worng Input\n");
        break;
    }
    fclose(fps);
    fclose(fpt);
    fclose(fpe);
    fclose(fpn);
    printf("If you want Exit Enter 0");
    scanf("%d",&cont);
}

}
else
{
    printf("Wrong Credential\n");
    loginregistrationmenu();
}
fclose(ptr);
}

void loginregistrationmenu(){
    int ch;
    printf("Enter 1 For Registration\n");
    printf("Enter 2 For Login\n");
    scanf("%d",&ch);
    printf("Press Enter to Continue");
    getch();
    system("cls");
    if (ch == 1)
    {

```

```

        registration();
    }
    else if(ch == 2){
        login();
    }
    else
    {
        printf("Wrong Input\n");
        printf("Press Enter to Continue");
        getch();
        system("cls");
        loginregistrationmenu();
    }
}

int mainmenu(){
    int choice;
    printf("Enter Your Choice\n");
    printf("Enter 1 For Students\n");
    printf("Enter 2 For Teachers\n");
    printf("Enter 3 For Staffs\n");
    printf("Enter 4 For Notice Board\n");
    printf("Enter 5 For Signout\n");
    scanf("%d",&choice);
    printf("Press Enter to Continue");
    getch();
    system("cls");
    return choice;
}

int studentmenu(){
    printf("Enter Your Choice\n");
    printf("Enter 1 For Add\n");
    printf("Enter 2 For View All Data\n");
    printf("Enter 3 For View Individually\n");
    printf("Enter 4 For Modify\n");
    printf("Enter 5 For Delete Recors\n");
    int ch;
    scanf("%d",&ch);
    printf("Press Enter to Continue");
    getch();
    system("cls");
    return ch;
}

int facultymenu(){
    printf("Enter Your Choice\n");
    printf("Enter 1 For Add\n");
    printf("Enter 2 For View All Data\n");
    printf("Enter 3 For View Individually\n");

```

```

        printf("Enter 4 For Modify\n");
        printf("Enter 5 For Delete Recors\n");
        int ch;
        scanf("%d",&ch);
        printf("Press Enter to Continue");
        getch();
        system("cls");
        return ch;
    }
    int staffmenu(){
        printf("Enter Your Choice\n");
        printf("Enter 1 For Add\n");
        printf("Enter 2 For View All Data\n");
        printf("Enter 3 For View Individually\n");
        printf("Enter 4 For Modify\n");
        printf("Enter 5 For Delete Recors\n");
        int ch;
        scanf("%d",&ch);
        printf("Press Enter to Continue");
        getch();
        system("cls");
        return ch;
    }

void addstudent(FILE * fp)
{
    char another='y';
    struct student s;
    fseek(fp,0,SEEK_END);
    while(another=='y' || another=='Y')
    {

        printf("\n\n\t\tEnter Full Name of Student: ");
        fflush(stdin);
        fgets(s.name,20,stdin);
        s.name[strlen(s.name)-1]='\0';

        printf("\n\n\t\tEnter Dept. Name: ");
        fflush(stdin);
        fgets(s.dept,20,stdin);
        s.dept[strlen(s.dept)-1]='\0';

        printf("\n\n\t\tEnter Adr. ");
        fflush(stdin);
        fgets(s.adress,20,stdin);
        s.adress[strlen(s.adress)-1]='\0';
        fflush(stdin);
        printf("\n\n\t\tEnter Roll number: ");
    }
}

```

```

        scanf("%d",&s.id);
        fflush(stdin);
        printf("\n\n\t\tEnter grade:");
        scanf("%f",&s.grade);

        fwrite(&s,sizeof(s),1,fp);
        printf("\n\n\t\tAdd another student?(Y/N)?");
        fflush(stdin);
        another=getchar();
        getch();
        printf("Press Enter to Continue");
        system("cls");
    }
}

void addfaculty(FILE * fp)
{
    char another='y';
    struct faculties f;
    fseek(fp,0,SEEK_END);
    while(another=='y' || another=='Y')
    {

        printf("\n\n\t\tEnter Full Name of Faculty: ");
        fflush(stdin);
        fgets(f.name,20,stdin);
        f.name[strlen(f.name)-1]='\0';

        printf("\n\n\t\tEnter Dept. Name: ");
        fflush(stdin);
        fgets(f.dept,20,stdin);
        f.dept[strlen(f.dept)-1]='\0';

        printf("\n\n\t\tEnter Adr. ");
        fflush(stdin);
        fgets(f.address,20,stdin);
        f.address[strlen(f.address)-1]='\0';
        fflush(stdin);
        printf("\n\n\t\tEnter ID: ");
        scanf("%d",&f.id);
        fflush(stdin);
        printf("\n\n\t\tEnter Contact Number:");
        fflush(stdin);
        fgets(f.number,20,stdin);

        fwrite(&f,sizeof(f),1,fp);
        printf("\n\n\t\tAdd another faculty Record?(Y/N)?");
        fflush(stdin);
        another=getchar();
    }
}

```

```

        getch();
        printf("Press Enter to Continue");
        system("cls");
    }
}

void addstaff(FILE * fp)
{
    char another='y';
    struct staffs sf;
    fseek(fp,0,SEEK_END);
    while(another=='y' || another=='Y')
    {

        printf("\n\n\t\tEnter Full Name of Staff: ");
        fflush(stdin);
        fgets(sf.name,20,stdin);
        sf.name[strlen(sf.name)-1]='\0';

        printf("\n\n\t\tEnter work Name: ");
        fflush(stdin);
        fgets(sf.dept,20,stdin);
        sf.dept[strlen(sf.dept)-1]='\0';

        printf("\n\n\t\tEnter Adr. ");
        fflush(stdin);
        fgets(sf.address,20,stdin);
        sf.address[strlen(sf.address)-1]='\0';
        fflush(stdin);
        printf("\n\n\t\tEnter ID number: ");
        scanf("%d",&sf.id);
        fflush(stdin);
        printf("\n\n\t\tEnter contact no.:");
        fflush(stdin);
        fgets(sf.number,20,stdin);
        fwrite(&sf,sizeof(sf),1,fp);
        printf("\n\n\t\tAdd another staff?(Y/N)?");
        fflush(stdin);
        another=getchar();
        getch();
        printf("Press Enter to Continue");
        system("cls");
    }
}

FILE * delstudent(FILE * fp)
{
    struct student s;
    int flag=0,tempRoll,siz=sizeof(s);

```

```

FILE *ft;

if((ft=fopen("temp.txt","wb+"))==NULL)
{
    printf("\n\n\t\t\t\t\t!!! ERROR !!!\n\n\t\t");
    system("pause");
    return fp;
}

printf("\n\n\tEnter Roll number of Student to Delete the Record");
printf("\n\n\t\t\t\t\tRoll No. : ");
scanf("%d",&tempRoll);

rewind(fp);

while((fread(&s,siz,1,fp))==1)
{
    if(s.id==tempRoll)
    {
        flag=1;
        printf("\n\tRecord Deleted for");
        printf("\n\n\t\t\t\t\t%s\n\n\t\t\t\t\t%s\n\n\t\t\t\t\t%d\n\n\t",s.name,s.dept,s
.id);
        continue;
    }

    fwrite(&s,siz,1,ft);
}

fclose(fp);
fclose(ft);

remove("studentdetails.txt");
rename("temp.txt","studentdetails.txt");

if((fp=fopen("stu.txt","rb+"))==NULL)
{
    printf("ERROR");
    return NULL;
}

if(flag==0) printf("\n\n\t\t\t\t\tNO STUDENT FOUND WITH THE INFORMATION\n\n\t\t");
printf("\n\n\t");
return fp;
}
FILE * delfaculty(FILE * fp)

```

```

{
    struct faculties f;
    int flag=0,tempRoll,siz=sizeof(f);
    FILE *ft;

    if((ft=fopen("temp.txt","wb+"))==NULL)
    {
        printf("\n\n\t\t\t\t\t!!! ERROR !!!\n\n\t\t");
        return fp;
    }

    printf("\n\n\tEnter ID number of Faculty to Delete the Record");
    printf("\n\n\t\t\t\t\tId No. : ");
    scanf("%d",&tempRoll);

    rewind(fp);

    while((fread(&f,siz,1,fp))==1)
    {
        if(f.id==tempRoll)
        {
            flag=1;
            printf("\n\tRecord Deleted for");
            printf("\n\n\t\t\t\t\t%s\n\n\t\t\t\t\t%s\n\n\t\t\t\t\t%d\n\t",f.name,f.dept,f
.id);
            continue;
        }

        fwrite(&f,siz,1,ft);
    }
    fclose(fp);
    fclose(ft);
    remove("facultydetails.txt");
    rename("temp.txt","facultydetails.txt");

    if((fp=fopen("facultydetails.txt","rb+"))==NULL)
    {
        printf("ERROR");
        return NULL;
    }

    if(flag==0) printf("\n\n\t\t\t\t\tNO Faculty FOUND WITH THE INFORMATION\n\n\t\t");
    return fp;
}
FILE * delstaff(FILE * fp)
{
    struct staffs sf;

```



```

int flag=0,tempRoll,siz=sizeof(sf);
FILE *ft;

if((ft=fopen("temp.txt","wb+"))==NULL)
{
    printf("\n\n\t\t\t!!! ERROR !!!\n\n\t\t");
    system("pause");
    return fp;
}

printf("\n\n\tEnter ID number of Staff to Delete the Record");
printf("\n\n\t\t\tId No. : ");
scanf("%d",&tempRoll);

rewind(fp);

while((fread(&sf,siz,1,fp))==1)
{
    if(sf.id==tempRoll)
    {
        flag=1;
        printf("\n\tRecord Deleted for");
        printf("\n\n\t\t\t%s\n\n\t\t\t%s\n\n\t\t\t%d\n\t",sf.name,sf.dept
,sf.id);
        continue;
    }

    fwrite(&sf,siz,1,ft);
}

fclose(fp);
fclose(ft);

remove("staffdetails.txt");
rename("temp.txt","staffdetails.txt");

if((fp=fopen("staffdetails.txt","rb+"))==NULL)
{
    printf("ERROR");
    return NULL;
}

if(flag==0) printf("\n\n\t\t\tNO STUDENT FOUND WITH THE INFORMATION\n\n\t\t");
return fp;
}

```

```

void modifyfaculty(FILE * fp)
{
    struct faculties f;
    int i,flag=0,tempRoll,siz=sizeof(f);

    printf("\n\n\tEnter ID Number of Faculty to MODIFY the Record : ");
    scanf("%d",&tempRoll);

    rewind(fp);

    while((fread(&f,siz,1,fp))==1)
    {
        if(f.id==tempRoll)
        {
            flag=1;
            break;
        }
    }

    if(flag==1)
    {
        fseek(fp,-siz,SEEK_CUR);
        printf("\n\n\t\t\tRecord Found\n\n\t\t\t");
        printf("\n\n\t\t\tFaculty Name: %s",f.name);
        printf("\n\n\t\t\tFaculty Roll: %d\n\n\t\t\t",f.id);
        printf("\n\n\t\t\tEnter New Data for the Faculty");
        printf("\n\n\t\t\tEnter Full Name of Faculty: ");
        fflush(stdin);
        fgets(f.name,100,stdin);
        f.name[strlen(f.name)-1]='\0';

        printf("\n\n\t\t\tEnter Department: ");
        fflush(stdin);
        fgets(f.dept,50,stdin);
        f.dept[strlen(f.dept)-1]='\0';

        printf("\n\n\t\t\tEnter Adr. ");
        fflush(stdin);
        fgets(f.address,20,stdin);
        f.address[strlen(f.address)-1]='\0';

        printf("\n\n\t\t\tEnter ID number: ");
        scanf("%d",&f.id);
        printf("\n\n\t\t\tEnter Contact no:\n");
        fflush(stdin);
        fgets(f.number,20,stdin);
        fwrite(&f,sizeof(f),1,fp);
    }
}

```

```

        else printf("\n\n\t!!!! ERROR !!!! RECORD NOT FOUND");
    }

void modifystaff(FILE * fp)
{
    struct staffs sf;
    int i,flag=0,tempRoll,siz=sizeof(sf);

    printf("\n\n\tEnter ID Number of Staff to MODIFY the Record : ");
    scanf("%d",&tempRoll);

    rewind(fp);

    while((fread(&sf,siz,1,fp))==1)
    {
        if(sf.id==tempRoll)
        {
            flag=1;
            break;
        }
    }

    if(flag==1)
    {
        fseek(fp,-siz,SEEK_CUR);
        printf("\n\n\t\t\tRecord Found\n\n\t\t\t");
        printf("\n\n\t\t\tStaff Name: %s",sf.name);
        printf("\n\n\t\t\tStaff Roll: %d\n\n\t\t\t",sf.id);
        printf("\n\n\t\t\tEnter New Data for the Staff");
        printf("\n\n\t\t\tEnter Full Name of Staff: ");
        fflush(stdin);
        fgets(sf.name,100,stdin);
        sf.name[strlen(sf.name)-1]='\0';
        printf("\n\n\t\t\tEnter Workname: ");
        fflush(stdin);
        fgets(sf.dept,50,stdin);
        sf.dept[strlen(sf.dept)-1]='\0';
        printf("\n\n\t\t\tEnter Adr. ");
        fflush(stdin);
        fgets(sf.address,20,stdin);
        sf.address[strlen(sf.address)-1]='\0';
        fflush(stdin);
        printf("\n\n\t\t\tEnter ID: ");
        scanf("%d",&sf.id);
        printf("\n\n\t\t\tEnter ContactNo:\n");
        fflush(stdin);
        fgets(sf.number,20,stdin);
    }
}

```

```

        fwrite(&sf, sizeof(sf), 1, fp);
    }

    else printf("\n\n\t!!!! ERROR !!!! RECORD NOT FOUND");
}

void modifystudent(FILE * fp)
{
    struct student s;
    int i, flag=0, tempRoll, siz=sizeof(s);

    printf("\n\n\tEnter Roll Number of Student to MODIFY the Record : ");
    scanf("%d", &tempRoll);

    rewind(fp);

    while((fread(&s, siz, 1, fp))!=1)
    {
        if(s.id==tempRoll)
        {
            flag=1;
            break;
        }
    }

    if(flag==1)
    {
        fseek(fp, -siz, SEEK_CUR);
        printf("\n\n\t\t\tRecord Found\n\n\t\t\t");
        printf("\n\n\t\t\tStudent Name: %s", s.name);
        printf("\n\n\t\t\tStudent Roll: %d\n\n\t\t\t", s.id);
        printf("\n\n\t\t\tEnter New Data for the student");

        printf("\n\n\t\t\tEnter Full Name of Student: ");
        fflush(stdin);
        fgets(s.name, 100, stdin);
        s.name[strlen(s.name)-1]='\0';

        printf("\n\n\t\t\tEnter Department: ");
        fflush(stdin);
        fgets(s.dept, 50, stdin);
        s.dept[strlen(s.dept)-1]='\0';

        printf("\n\n\t\t\tEnter Adr. ");
        fflush(stdin);
        fgets(s.address, 20, stdin);
        s.address[strlen(s.address)-1]='\0';
    }
}

```

```

        fflush(stdin);
        printf("\n\n\t\tEnter Roll number: ");
        scanf("%d",&s.id);
        fflush(stdin);
        printf("\n\n\t\tEnter Grade\n");
        scanf("%f",&s.grade);
        fwrite(&s,sizeof(s),1,fp);
    }

    else printf("\n\n\t!!!! ERROR !!!! RECORD NOT FOUND");
}

void displaystudent(FILE * fp)
{
    struct student s;
    int i,siz=sizeof(s);

    rewind(fp);

    while((fread(&s,siz,1,fp))==1)
    {
        printf("\n\t\tNAME : %s",s.name);
        printf("\n\n\t\tDepartment : %s",s.dept);
        printf("\n\n\t\tAdr. : %s",s.address);
        printf("\n\n\t\tROLL : %d",s.id);
        printf("\n\n\tGrade: %f\n",s.grade);
    }
}

void displayfaculty(FILE * fp)
{
    struct faculties f;
    int i,siz=sizeof(f);

    rewind(fp);

    while((fread(&f,siz,1,fp))==1)
    {
        printf("\n\t\tNAME : %s",f.name);
        printf("\n\n\t\tDepartment : %s",f.dept);
        printf("\n\n\t\tAdr. : %s",f.adress);
        printf("\n\n\t\tID : %d",f.id);
        printf("\n\n\tNumber: %s\n",f.number);
    }
}

void displaystaff(FILE * fp)
{
    struct staffs sf;
    int i,siz=sizeof(sf);

```

```

rewind(fp);

while((fread(&sf,siz,1,fp))==1)
{
    printf("\n\t\tNAME : %s",sf.name);
    printf("\n\n\t\tDepartment : %s",sf.dept);
    printf("\n\n\t\tAdr. : %s",sf.address);
    printf("\n\n\t\tID : %d",sf.id);
    printf("\n\n\t\tNumber: %s\n",sf.number);
}
}

void Individualstudent(FILE *fp)
{
    int tempRoll,flag,siz,i;
    struct student s;
    char another='y';

    siz=sizeof(s);

    while(another=='y' || another=='Y')
    {
        printf("\n\n\tEnter Roll Number: ");
        scanf("%d",&tempRoll);

        rewind(fp);

        while((fread(&s,siz,1,fp))==1)
        {
            if(s.id==tempRoll)
            {
                flag=1;
                break;
            }
        }

        if(flag==1)
        {
            printf("\n\t\tNAME : %s",s.name);
            printf("\n\n\t\tDepartment : %s",s.dept);
            printf("\n\n\t\tAddress : %s",s.address);
            printf("\n\n\t\tROLL : %d",s.id);
            printf("\n\n\t\tGrade : %.2f\n\t",s.grade);
        }
        else printf("\n\n\t\t!!!! ERROR RECORD NOT FOUND !!!!!");
    }
}

```

```

        printf("\n\n\t\tShow another student information? (Y/N)?");
        fflush(stdin);
        another=getchar();
    }
}
void Individualfaculty(FILE *fp)
{
    int tempRoll,flag,siz,i;
    struct faculties f;
    char another='y';

    siz=sizeof(f);

    while(another=='y' || another=='Y')
    {
        printf("\n\n\tEnter ID Number: ");
        scanf("%d",&tempRoll);

        rewind(fp);

        while((fread(&f,siz,1,fp))==1)
        {
            if(f.id==tempRoll)
            {
                flag=1;
                break;
            }
        }

        if(flag==1)
        {
            printf("\n\t\tNAME : %s",f.name);
            printf("\n\n\t\tDepartment : %s",f.dept);
            printf("\n\n\t\tAdress : %s",f.adress);
            printf("\n\n\t\tID : %d",f.id);
            printf("\n\n\t\tContact Number : %s\n\t",f.number);

        }
        else printf("\n\n\t\t!!!! ERROR RECORD NOT FOUND !!!!!");

        printf("\n\n\t\tShow another student information? (Y/N)?");
        fflush(stdin);
        another=getchar();
    }
}
void Individualstaff(FILE *fp)
{
    int tempRoll,flag,siz,i;

```

```

struct staffs sf;
char another='y';

siz=sizeof(sf);

while(another=='y' || another=='Y')
{
    printf("\n\n\tEnter ID Number: ");
    scanf("%d",&tempRoll);

    rewind(fp);

    while((fread(&sf,siz,1,fp))==1)
    {
        if(sf.id==tempRoll)
        {
            flag=1;
            break;
        }
    }

    if(flag==1)
    {
        printf("\n\t\tNAME : %s",sf.name);
        printf("\n\n\t\tDepartment : %s",sf.dept);
        printf("\n\n\t\tAddress : %s",sf.address);
        printf("\n\n\t\tID : %d",sf.id);
        printf("\n\n\t\tContact Number : %s\n\t",sf.number);

    }
    else printf("\n\n\t\t!!!! ERROR RECORD NOT FOUND !!!!!");

    printf("\n\n\t\tShow another student information? (Y/N)?");
    fflush(stdin);
    another=getchar();
}
}

int noticeboard(){
    int ch;
    printf("Enter Your Choice:\n");
    printf("1. For Adding The Notice Board Enter 1\n");
    printf("2. For Viewing The Notice Board Enter 2\n");
    printf("3. For Editing The Notice Board Enter 3\n");
    scanf("%d",&ch);
    return ch;
}

void addnotice(FILE *fp){
    char another='y';

```



```

    struct notice n;
    fseek(fp,0,SEEK_END);
    while(another=='y' || another=='Y')
    {

        printf("\n\n\t\tEnter Notice: ");
        fflush(stdin);
        fgets(n.details,200,stdin);
        n.details[strlen(n.details)-1]='\0';
        fflush(stdin);
        printf("\n\n\t\tEnter SL. No.\n");
        scanf("%f",&n.slno);

        fwrite(&n,sizeof(n),1,fp);

        printf("\n\n\t\tAdd another Notice?(Y/N)?");
        fflush(stdin);
        another=getchar();
    }
}

void viewnotice(FILE *fp){
    struct notice n;
    int i,siz=sizeof(n);

    rewind(fp);

    while((fread(&n,siz,1,fp))==1)
    {
        printf("\n\t\tNOTICE : %s",n.details);
        printf("\n\t\tSL.No : %s",n.slno);
    }
}

void editnotice(FILE *fp){
    struct notice n;
    int i,flag=0,tempRoll,siz=sizeof(n);

    printf("\n\n\t\tEnter SL. Number of Notice to MODIFY the Record : ");
    scanf("%d",&tempRoll);
    rewind(fp);
    while((fread(&n,siz,1,fp))==1)
    {
        if(n.slno==tempRoll)
        {
            flag=1;
            break;
        }
    }
}

```

```

if(flag==1)
{
    fseek(fp,-siz,SEEK_CUR);
    printf("\n\n\t\t\tRecord Found\n\n\t\t\t");
    printf("\n\n\t\t\tNotice Details: %s",n.details);
    printf("\n\n\t\t\tSL.No: %d\n\n\t\t\t",n.slno);
    printf("\n\n\t\t\tEnter New Data for the Notice");

    printf("\n\n\t\t\tEnter The Notice: ");
    fflush(stdin);
    fgets(n.details,100,stdin);
    n.details[strlen(n.details)-1]='\0';
    printf("\n\n\t\t\tEnter SLNO.\n");
    scanf("%f",&n.slno);
    fwrite(&n,sizeof(n),1,fp);
}

else {

    printf("\n\n\t!!!! ERROR !!!! RECORD NOT FOUND");

}
}

```

Sample Outputs

1. Register And Login Menu:

```

PS C:\Users\SHUBHASAI\Desktop\university management> cd "c:\Users\SHUBHASAI\Desktop\university management\" ; if ($?) { gcc main.c -o main } ; if ($?) { .\main }
Enter 1 For Registration
Enter 2 For Login
1

```

2. Registration and Login:

```

Enter Username
shubhasai
Create Password
shubhasai123
Successfully Register

```

```

Enter User Name:
shubhasai
Enter password:
shubhasai123
Press Enter to Continue

```

3. Main Menu:

```

WELCOME
Enter Your Choice
Enter 1 For Students
Enter 2 For Teachers
Enter 3 For Staffs
Enter 4 For Notice Board
Enter 5 For Changing password

```

4. Student Menu:

```

Welcome to Student Section:
Enter Your Choice
Enter 1 For Add
Enter 2 For View All Data
Enter 3 For View Individually
Enter 4 For Modify
Enter 5 For Delete Recors

```

5. Functionalities:

```

Enter Full Name of Student: SRIPATI PANDEY

Enter Dept. Name: ETC

Enter Adr. BIHAR

Enter Roll number: 52

Enter grade:8.0

Add another student?(Y/N)?

```

```

Enter Roll number of Student to Delete the Record

Roll No. : 53

Record Deleted for

SHUBHASAI MOHAPATR

ETC

53

ERRORIf you want Exit Enter 0

```

```

Enter Roll Number: 52

NAME : SRIPATI PANDEY
Department : ETC
Adress : BIHAR
ROLL : 52

SGPA:

Grade : 8.00

Show another student information? (Y/N)?

```

```

NAME : SHUBHASAI MOHAPATR
Department : ETC
Adr. : BHADRAK, 7205074901
ROLL : 53
Grade: 7.500000

NAME : SRIPATI PANDEY
Department : ETC
Adr. : BIHAR
ROLL : 52
Grade: 8.000000

NAME : Shivanshu Singh
Department : ETC
Adr. : NA
ROLL : 51
Grade: 8.530000
If you want Exit Enter 0

```

Contribution by group members

B220051: Menu Design

B220052: Login and registration details of User

B220053: File Handling

References

Websites:

StackOverflow: <https://stackoverflow.com>

Geeks For Geeks: <https://www.geeksforgeeks.org>

Programiz: <https://www.programiz.com/>

Github Repo: https://github.com/shubhasai/University_Management.git