

SCHOOL BILLING SYSYTEM

A Mini Project

Academic Year: 2021 - 22 ODD SEMESTER

**Department with Specialization : Computer Science and
Engineering with Artificial
Intelligence and Machine
Learning**

Semester : I
Course Code : 18CSS101J
**Course Title : Programming and Problem
Solving**

Submitted by

**ROHAN KOLLIPARA[RA2111026010263]
RATHNA SEKHAR MAKKENA[RA2111026010264]**

Under the Guidance of

Dr. A. Suresh
(Associate Professor, NWC)

**DEPARTMENT OF
COLLEGE OF
TECHNOLOGY**



SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

**COMPUTING
ENGINEERING AND**

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

JANUARY 2022

AIM AND ABSTRACT

AIM: To create the “School Billing System” using the C Programming Language

ABSTRACT

This project “School Billing System” helps many of the institutions for keeping a record of the student information and student fee information. This application is friendly to the users and it is quite easy to understand to how to record or store the information in the application. This School Billing System application is developed by using the C language. This is an error-free application console and the user could not face any difficulty while using this application.

Achieving this job by a person is not an easy task as the information is scattered and it is exceedingly difficult to find a particular student or a teacher information, and this could be a time consuming. With the help of this application many people can save time and maintain a detailed information and can be save the information for a long time.

ALGORITHM

➤ STEP 1- Start

➤ STEP 2-Ask the user to enter the account type i.e., whether he/she want to choose

1)Student

2) Teacher And Staff

3) Exit

➤ STEP 3-If the user selects student, then he/she gets six options, they are

1)Add student record

- 2)Select student record
- 3)Modify student record
- 4)Delete student record
- 5)Exit

Choose the required option do you want

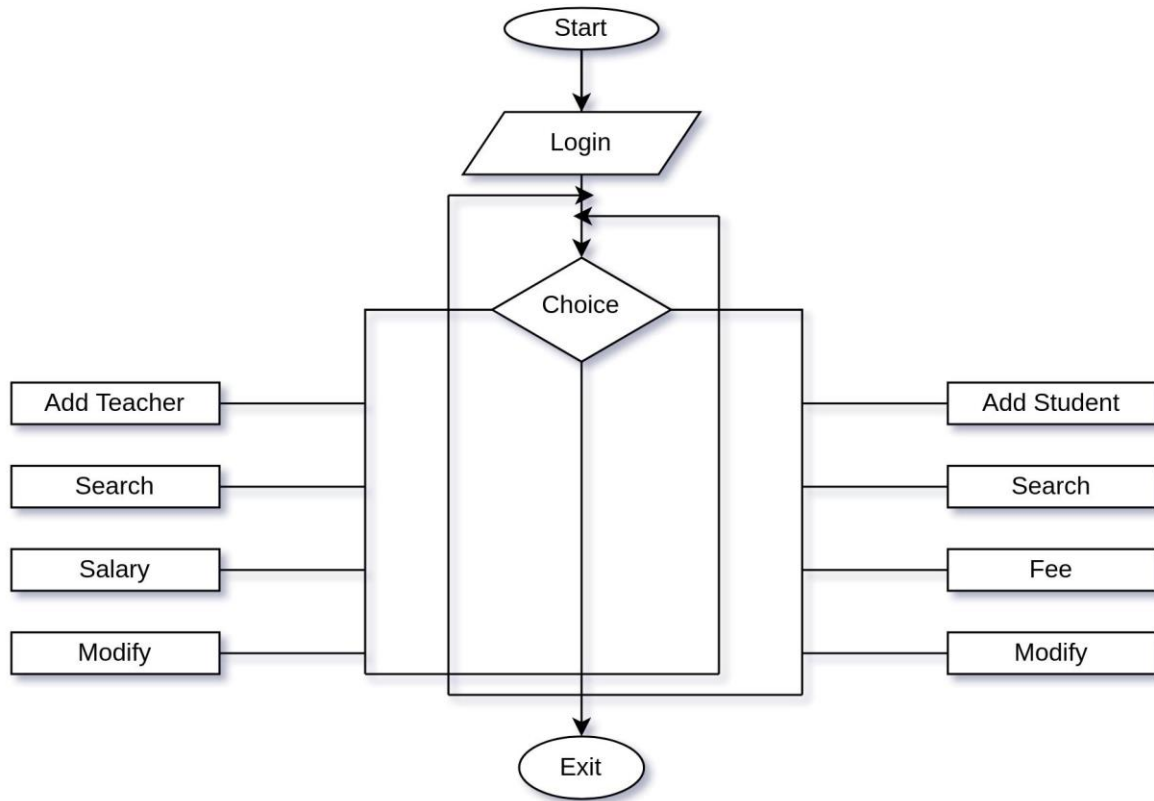
- STEP 4- If the user chooses teachers and staff then he/she will get six options, they are

- 1)Add Staff record
- 2)Search Staffs record
- 3)Modify Staffs record
- 4)Delete staff record
- 5)Calculate staff salary
- 6)Exit

Choose the required option do you want

- STEP 5- If the user chooses option exit, he/she will totally exit from the school billing program
- STEP 6- Stop

FLOW CHART



SOURCE CODE

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<conio.h>
#include<windows.h>
struct date_and_time

{
    int dy,mth;

};
int check_class();//check class (1-12)
struct student
{
    struct date_and_time dte_tme;
    float fes,debt,total,advanced,duedate;
    char names[50];
    int ro,cl;
} std,s;
struct teacher
{
    struct date_and_time dte_tme;
    char names[50];
    float slry,advanced,total;
    int t_id,t_no;
} techr,temp_f;
int checking_date(int,int);// for checking date
```

```
void adding_records(int); //for adding records
void update_records(int); //for modifying records
void searching_records(int); //for searching records
void deleting_records(int); //for deleting records
void salary(int); //for the calculation of salary of teacher and staff
FILE *file_store, *file_temp; //file declaration

int mm, dd; //mm=month, dd=day

void exits(); //for exiting

void main(void)
{
    int l, m, o;

    printf("\n\t\t\t#####\n");
    printf("\n\t\t\t|\n");
    printf("\n\t\t\t**WELCOME TO SRM SCHOOL BILLING**\n");
    printf("\n\t\t\t|\n");
    printf("\n\t\t\t#####\n");

    printf("\n\tPLEASE ENTER ANY KEY TO CONTINUE");
    for(l=0; l<5; l++)
    {
        printf(".");
        Sleep(500); //after printing one . another comes after 0.5 seconds
    }
    getch();
    system("cls"); //clears the screen
    printf("\n");
```

```

printf("\n\n\t\t\t **WELCOME TO SRM SCHOOL BILLING**\n\n\n");

printf("\nPLEASE ENTER ANY KEY TO START\n");
for(l=0; l<5; l++)
{
    printf(".");
    Sleep(500);
}
fflush(stdin);
getch();
system("cls");
system("color 0f");//1st is for back ground color and second is for text color
printf("\n\n\t\t PLEASE ENTER CURRENT DAY=");
scanf("%d",&dd);
printf("\n\n\t\t PLEASE ENTER CURRENT MONTH=");
scanf("%d",&mm);
mm=checking_date(mm,dd);
start();
}

void start()
{
    int l,m;//j is for selection of account type
    system("cls");
    printf("\n\n\t\t PLEASE ENTER ACCOUNT TYPE");
    printf("\n\n\t\t1:: Student");
    printf("\n\n\t\t2:: Teachers and Staffs");
    printf("\n\n\t\t3:: Exit");

```

```

printf("\n\t\tAccount Type Choice ");
fflush(stdin);
scanf("%d",&m);
switch (m)
{
case 3:
    exits();
case 1:
{
    system("cls");
    printf("\n\t\tPLEASE ENTER THE CHOICE");
    printf("\n\t\t1:: Add Record");
    printf("\n\t\t2:: Search Record");
    printf("\n\t\t3:: Modify Record");
    printf("\n\t\t4:: Delete Record");
    printf("\n\t\t5:: Calculate Fee");
    printf("\n\t\t6:: Exit");
    printf("\n\n Enter Choice ");
    fflush(stdin);
    scanf("%d",&l);
    switch (l)
    {
case 1:
        adding_records(m);//function call
        start();//function call
case 2:
        searching_records(m);
        start();
case 3:

```



```

        update_records(m);

        start();
case 4:
        deleting_records(m);

        start();
case 5:
        fee(mm);

        start();
case 6:
        exits();
default :
{
        printf("\n\n\tInvalid Entry!!");
        printf("\n\n\tTo Account Type\n\n\t");
        system("Pause");
        start();
}
}
}
case 2:
{

        system("cls");
        printf("\n\n\tPLEASE ENTER THE CHOICE");
        printf("\n\n\t1:: Add Record");
        printf("\n\n\t2:: Search Record");
        printf("\n\n\t3:: Modify Record");
        printf("\n\n\t4:: Delete Record");
        printf("\n\n\t5:: Calculate Salary");

```

```
printf("\n\t\t6:: Exit");  
printf("\n\n Enter Choice ");  
fflush(stdin);  
scanf("%d",&l);  
switch (l)  
{  
case 1:  
    adding_records(m);  
    start();  
case 2:  
    searching_records(m);  
    start();  
case 3:  
    update_records(m);  
    start();  
case 4:  
    deleting_records(m);  
case 5:  
    salary(mm);  
    start();  
case 6:  
    exits();  
default :  
{  
    printf("\n\n\tInvalid entry!!");  
    printf("\n\nTo Account Type\n\n\t");  
    system("Pause");  
    start();  
}
```

```

    }
}
default :
{
    printf("\n\n\tInvalid Entry!!");
    printf("\n\nTo Account Type\n\n\t");
    system("Pause");
    start();
}
}
}

```

```

void adding_records(int j)
{
    int df,clrdate,duedte,mh=0;//cdat=month till which fee is cleared
    float fee_fee;//used in calculatin of fee of different class
    char cs='y';
    system("cls");
    printf("\n\t*****");

    printf("\n\t          *****          ");

    printf("\n\t*****          ADD RECORD          *****");

    printf("\n\t          *****          ");

    printf("\n\t*****");
    if (j==1)

```

```

{
while(cs=='y' | |cs=='Y')
{
    int a=1;

    printf("\n\nEnter the name of student: ");

    fflush(stdin);

    scanf("%[^\n]",std.names);

    printf("\n\nEnter the class: ");

    fflush(stdin);

    std.cl=check_class();

    printf("\n\nEnter the Roll No.:");

    fflush(stdin);

    scanf("%2d",&std.ro);

    printf("\n\nEnter day and month till which fee is paid:");

    fflush(stdin);

    scanf("%2d%2d",&duedte,&clrdate);

    clrdate=checking_date(clrdate,duedte);

    std.dte_tme.mth=clrdate;

    fee_fee=std.cl/10.0;

    std.fes=1000*(1+fee_fee);//fee of different classes

    df=mm-std.dte_tme.mth;//months of fee left to be paid

    std.debt=(df*std.fes)*1/100;

    std.duedate=(df)*std.fes;//fees left to be paid

    if(df==1)
    {

        std.total=std.fes;

        std.debt=0;

    }

    else

```

```

{
    std.total=std.debt+std.duedate;
}

//for calculation of total fee

file_store=fopen("student","ab+");//opening a binary file in apend mode
fwrite(&std,sizeof(std),1,file_store);
fclose(file_store);

printf("\n\nDo you want to continue with the process(press y or n");
fflush(stdin);

cs=getch();
}

getch();
}

if (j==2)
{
    while(cs=='y' | cs=='Y')
    {
        int a=1;

        printf("\n\nEnter name of teacher/staff:" );
        fflush(stdin);

        scanf("%[^\\n]",techr.names);

        printf("\n\nEnter teacher/staff id: ");
        fflush(stdin);

        scanf("%d",&techr.t_id);

        printf("\n\nEnter number of class/shift per month:: ");
        scanf("%d",&techr.t_no);
        fflush(stdin);

        printf("\n\nEnter  day and month till which salary is paid::");
        scanf("%d %d",&techr.dte_tme.dy,&techr.dte_tme.mth);

        clrdate=checking_date(duedte,clrdate);
    }
}

```

```

    techr.dte_tme.mth=clrddate;
    techr.slry=techr.t_no*500;
    techr.advanced=(techr.dte_tme.mth-mm-1)*techr.slry;
    if (techr.advanced<0) techr.advanced=0;
    techr.total=techr.slry;
    file_temp=fopen("teacher","ab+");
    fwrite(&techr,sizeof(techr),1,file_temp);
    fclose(file_temp);
    printf("\n\nDo you want to continue with the process(press y or n");
    fflush(stdin);
    cs=getch();
}
fflush(stdin);
printf("\n\n");
system("pause");
}
}

```

```

void searching_records(int j)
{
    char name_s[50],name_t[50];
    int p=1,option;
    char cs='y';
    if (j==1)
    {
        while(cs=='y' || cs=='Y')
        {
            int p=1;
            system("cls");

```

```
printf("\n\t*****");
```

```
printf("\n\t*****");
```

```
printf("\n\t***** SEARCH RECORD *****");
```

```
printf("\n\t*****");
```

```
printf("\n\t*****");
```

```
printf("\n\n\t\tPLEASE CHOOSE SEARCH TYPE::");
```

```
printf("\n\n\t\t1::Search by Name::");
```

```
printf("\n\n\t\t2::Search by Class::");
```

```
printf("\n\n\t\t3::Exit");
```

```
printf("\n\n\t\t::Enter your Choice:: ");
```

```
fflush(stdin);
```

```
scanf("%d",&option);
```

```
if (option==1)
```

```
{
```

```
    p=1;
```

```
    printf("\n\nEnter name of student to search: ");
```

```
    fflush(stdin);
```

```
    scanf("%[^\n]",name_s);
```

```
    file_store=fopen("student","r");
```

```
    while(fread(&std,sizeof(std),1,file_store)==1)
```

```
    {
```

```
        if (strcmpi(name_s,std.names)==0)
```

```
        {
```

```
            p=0;
```

```
            printf("\nname = %s",std.names);
```

```

        printf("\n class = %d",std.cl);
        printf("\n roll no = %d",std.ro);
        printf("\n monthly fee =%.2f",std.fes);
        printf("\n last fee paid in month =%2d",std.dte_tme.mth);
        printf("\n due= %.2f",std.duedate);
        printf("\n fine= %.2f",std.debt);
        printf("\n total= %.2f\n\n",std.total);
    }
}

if (p==1)
    printf("\n\n RECORD NOT FOUND");
printf("\n\n");
system("pause");
fflush(stdin);
fclose(file_store);
}

else if (option==2)
{
    int cl;

    p=1;
    printf("\n\n Enter class of student to search: ");
    fflush(stdin);
    cl=check_class();
    file_store=fopen("student","rb");
    while(fread(&std,sizeof(std),1,file_store)==1)
    {
        if (std.cl==cl)
        {
            p=0;

```



```

        printf("\nname = %s",std.names);
        printf("\nclass = %d",std.cl);
        printf("\nroll no = %d",std.ro);
        printf("\nmonthly fee =%.2f",std.fes);
        printf("\nlast fee paid in month =%2d",std.dte_tme.mth);
        printf("\n due=%.2f",std.duedate);
        printf("\n fine=%.2f",std.debt);
        printf("\n total=%.2f",std.total);
    }
}

if (p==1)
    printf("\n\nRECORD NOT FOUND");
    printf("\n\n");
    system("pause");
    fflush(stdin);
    fclose(file_store);
}

else if(option==3)
{
    exits();
}
else
{
    printf("\n\n\n\t\tINVALID ENTRY!!!!\n\n\t\t");
    system("pause");
    searching_records(1);
}

printf("\n\nDo you want to continue with the process(press y or n");

```

```

        fflush(stdin);

        cs=getch();
    }

    getch();
}

if (j==2)
{
    while(cs=='y' || cs=='Y')
    {
        int p=1;

        printf("\n\nname of teacher/staff to search: ");

        fflush(stdin);

        scanf("%[^\\n]",name_t);

        file_temp=fopen("teacher","rb");

        while(fread(&techr,sizeof(techr),1,file_temp)==1)
        {
            if (strcmp(name_t,techr.names)==0)
            {
                p=0;

                printf("\nname = %s",techr.names);

                printf("\nteacher/staff id = %d",techr.t_id);

                printf("\nmonth till when salary is paid =%d",techr.dte_tme.mth);

                printf("\nmonthly salary = %.2f",techr.slry);

                printf("\nadvance paid = %.2f",techr.advanced);
            }
        }
    }

    if (p==1)

        printf("\n\nRECORD NOT FOUND");

    printf("\n\n");
}

```

```

        system("pause");
        fflush(stdin);
        fclose(file_temp);
        printf("\n\nDo you want to continue with the process(press y or n");
        fflush(stdin);
        cs=getch();
    }
    getch();
}

}

void update_records(int j)
{

    char name_s[50];
    int p=1,option,cll,roll_no;
    char cs='y';
    if (j==1)
    {
        while(cs=='y' | cs=='Y')
        {
            system("cls");
            printf("\n\t*****");

            printf("\n\t          *****          ");

            printf("\n\t*****  MODIFY RECORD  *****");

```

```

printf("\n\t          *****          ");

printf("\n\t*****");

printf("\n\n\tPLEASE CHOOSE MODIFY TYPE::");

printf("\n\n\t\t1::Modify by Name::");

printf("\n\n\t\t2::Modify by Name &Class::");

printf("\n\n\t\t3::Modify by Name,Class & Roll No::");

printf("\n\n\t\t4::Exit");

printf("\n\n\t\t::Enter your Choice:: ");

fflush(stdin);

scanf("%d",&option);

if (option==1)
{
    int p=0;

    printf("\n\nEnter Name of Student to Modify: ");

    fflush(stdin);

    scanf("%s",name_s);

    file_store=fopen("student","rb+");

    while(fread(&std,sizeof(std),1,file_store)==1)
    {
        p=1;

        if (strcmp(name_s,std.names)==0)
        {
            p=0;

            printf("\nEnter New Name of Student: ");

            fflush(stdin);

            scanf("%s",std.names);

            printf("\nEnter New Class of Student: ");

            fflush(stdin);

```

```

        std.cl=check_class();

        printf("\nEnter New Roll of Student: ");

        fflush(stdin);

        scanf("%d",&std.ro);

        fseek(file_store,-sizeof(std),SEEK_CUR);

        fwrite(&std,sizeof(std),1,file_store);

        fclose(file_store);

    }

}

if (p==1)

    printf("\n\nRECORDS NOT FOUND");

else

    printf("\n\nRECORDS SUCCESSFULLY MODIFIED");

    printf("\n\n");

    system("pause");

}

else if (option==2)

{

    int p=0;

    printf("\n\nEnter Name of Student to Modify: ");

    fflush(stdin);

    scanf("%[^\\n]",name_s);

    printf("\n\nEnter Class of Student to Modify: ");

    fflush(stdin);

    cll=check_class();

    file_store=fopen("student","rb+");

    while(fread(&std,sizeof(std),1,file_store)==1)

    {

        p=1;
    }
}

```

```

        if (strcmp(name_s,std.names)==0 && cll==std.cl)
        {
            p=0;

            printf("\nEnter New Name of Student: ");

            fflush(stdin);

            scanf("%[^\\n]",std.names);

            printf("\nEnter New Class of Student: ");

            fflush(stdin);

            std.cl=check_class();

            printf("\nEnter New Roll of Student: ");

            fflush(stdin);

            scanf("%d",&std.ro);

            fseek(file_store,-sizeof(std),SEEK_CUR);

            fwrite(&std,sizeof(std),1,file_store);

            fclose(file_store);

        }
    }

    if (p==1)

        printf("\n\nRECORDS NOT FOUND");

    else

        printf("\n\nRECORDS SUCCESSFULLY MODIFIED");

    printf("\n\n");

    system("pause");

}

else if (option==3)

{

    int p=0;

    printf("\n\nenter name of student to modify: ");

    fflush(stdin);

```

```

scanf("%[^\\n]",name_s);

printf("\\n\\nenter class of student to modify: ");

fflush(stdin);

c11=check_class();

printf("\\n\\nenter roll of student to modify: ");

fflush(stdin);

scanf("%d",&roll_no);

file_store=fopen("student","rb+");

while(fread(&std,sizeof(std),1,file_store)==1)
{
    p=1;

    if (strcmpi(name_s,std.names)==0 && c11==std.cl && roll_no==std.ro)
    {
        p=0;

        printf("\\nenter new name of student: ");

        fflush(stdin);

        scanf("%[^\\n]",std.names);

        printf("\\nenter new class of student: ");

        fflush(stdin);

        std.cl=check_class();

        printf("\\nenter new roll of student: ");

        fflush(stdin);

        scanf("%d",&std.ro);

        fseek(file_store,-sizeof(std),SEEK_CUR);

        fwrite(&std,sizeof(std),1,file_store);

        fclose(file_store);
    }
}

if (p==1)

```

```

        printf("\n\nRECORDS NOT FOUND");
    else
        printf("\n\nRECORDS SUCCESSFULLY  MODIFIED");
    printf("\n\n");
    system("pause");
}
else if (option==4) exits();
else
{
    printf("\n\n\n\t\tINVALID ENTRY!!!!\n\n\t\t");
    system("pause");
    update_records(1);
}

printf("\n\nDo you want to continue with the process(press y or n");
fflush(stdin);
cs=getch();
}
getch();
}

if (j==2)
{
    while(cs=='y' | |cs=='Y')
    {
        int p=1;
        printf("enter name of teacher to modify: ");
        fflush(stdin);
        scanf("%[^\\n]",name_s);
    }
}

```



```

file_temp=fopen("teacher","rb+");
while(fread(&techr,sizeof(techr),1,file_temp)==1)
{
    if (strcmpi(name_s,techr.names)==0)
    {
        p=0;
        printf("\nenter new name of teacher: ");
        fflush(stdin);
        scanf("%s",techr.names);
        printf("\nenter new id of teacher: ");
        fflush(stdin);
        scanf("%d",&techr.t_id);
        fseek(file_temp,-sizeof(techr),SEEK_CUR);
        fwrite(&techr,sizeof(techr),1,file_temp);
        fclose(file_temp);
    }
}

if (p==1)
    printf("\n\nRECORD NOT FOUND");
else
    printf("\n\nRECORD SUCCESSFULLY  MODIFIED");
printf("\n\n");
system("pause");
fflush(stdin);

printf("\n\nDo you want to continue with the process(press y or n");
fflush(stdin);
cs=getch();

```

```

    }

    getch();
}
}

void deleting_records(int j)
{
    system("cls");

    printf("\n\t*****");

    printf("\n\t          *****");

    printf("\n\t*****          DELETE RECORD          *****");

    printf("\n\t          *****");

    printf("\n\t*****");

    FILE *temporary,*tmp1;

    int p=1;

    char name_s[50],cs='y';

    if (j==1)
    {
        while(cs=='y' | cs=='Y')
        {
            int p=1;

            printf("\n\tnenter name of student to delete: ");

            fflush(stdin);

            scanf("%[^\n]",name_s);

            file_store=fopen("student","rb");

```

```
temporary=fopen("tempfile","wb");//opening of temporary file for deleting process
```

```
while (fread(&std,sizeof(std),1,file_store)==1)
```

```
{
```

```
    if (strcmp(std.names,name_s)==0)
```

```
    {
```

```
        p=0;
```

```
        continue;
```

```
    }
```

```
    else
```

```
    {
```

```
        fwrite(&std,sizeof(std),1,temporary);
```

```
    }
```

```
}
```

```
if (p==1)
```

```
    printf("\n\nRECORD NOT FOUND");
```

```
else
```

```
    printf("\n\nRECORD SUCCESSFULLY DELETED");
```

```
printf("\n\n");
```

```
system("pause");
```

```
fflush(stdin);
```

```
fclose(file_store);
```

```
fclose(temporary);
```

```
system("del student");/*all data except the data to be
```

```
deleted in student were 1st moved to temp and data in student
```

```
was deleted*/
```

```
system("ren tempfile, student");//renaming temp to student
```

```
printf("\n\nDo you want to continue with the process(press y or n");
```

```

        fflush(stdin);
        cs=getch();
    }
    getch();
}

if (j==2)
{
    p=1;
    char name_t[50];
    while(cs=='y' | |cs=='Y')
    {
        printf("\n\nEnter name of teacher to delete record: ");
        fflush(stdin);
        scanf("%[^\\n]",name_t);
        file_temp=fopen("teacher","rb");
        tmp1=fopen("tempfile1","wb");
        while (fread(&techr,sizeof(techr),1,file_temp)==1)
        {
            if (strcmp(techr.names,name_t)==0)
            {
                p=0;
                continue;
            }
            else
            {
                fwrite(&techr,sizeof(techr),1,tmp1);
            }
        }
    }
}

```

```

    if (p==1)
        printf("\n\nRECORD NOT FOUND");
    else
        printf("\n\nRECORD SUCCESSFULLY DELETED");
    printf("\n\n");
    system("pause");
    fflush(stdin);

    fclose(file_temp);
    fclose(tmp1);
    system("del teacher");
    system("ren tempfile1, teacher");
    printf("\n\nDo you want to continue with the process(press y or n");
    fflush(stdin);
    cs=getch();
}
getch();
}

void salary(int mm)
{
    system("cls");
    printf("\n\t*****");

    printf("\n\t\t\t\t\t*****\t\t\t\t\t");

    printf("\n\t*****\t\t\t\t\tSALARY\t\t\t\t\t*****");

```

```

printf("\n\t *****");

printf("\n\t*****");
FILE *files,*temp_f;
int p=1,day_s;
char name_s[50],cs='y';
int months,df,ids;
while(cs=='y' || cs=='Y')
{
    int p=1;
    fflush(stdin);
    printf("\n\nEnter name:: ");
    scanf("%[^\n]",name_s);
    printf("\n\nEnter ID:: ");
    scanf("%d",&ids);
    files=fopen("teacher","rb+");
    temp_f=fopen("te","wb+");
    while(fread(&techr,sizeof(techr),1,files)==1)//file opened
    {
        if(strcmp(techr.names,name_s)==0)//name entered is compared to the existing name in file
        {
            float lsal;
            p=0;
            printf("\n\nEnter the Day & months till which salary is to be paid:: ");
            fflush(stdin);
            scanf("%d%d",&day_s,&months);
            months=checking_date(day_s,months);
            techr.advanced=(months-mm-1)*techr.slry;
            if (techr.advanced<0) techr.advanced=0;

```

```

lsal=mm-techr.dte_tme.mth;//months of salary left to be paid
if(lsal<0) lsal=0;
techr.total=techr.advanced+techr.slry*(1+lsal);
if(months==techr.dte_tme.mth) techr.total=0;
printf("\nmonthly salary left to be paid:: %.2f",lsal);
printf("\ntotal :: %.2f",techr.total);
printf("\nadvance :: %.2f",techr.advanced);
techr.dte_tme.mth=months;
fwrite(&techr,sizeof(techr),1,temp_f);
fclose(files);
fclose(temp_f);

if (p==1)
    printf("\n\nRECORD NOT FOUND");
printf("\n\n");
system("pause");
fflush(stdin);
system("del teacher");
system("ren te, teacher");
}
}
printf("\n\nDo you want to continue with the process(press y or n");
fflush(stdin);
cs=getch();
}
getch();
}

```

```

void fee(int mm)

```

```

{
system("cls");

printf("\n\t*****");

printf("\n\t          *****          ");

printf("\n\t*****          FEE          *****");

printf("\n\t          *****          ");

printf("\n\t*****");
FILE *files,*temp_f;
int p=0;
char name_s[50],cs='y';
int cls, rollss,monthss,diffs;
while(cs=='y' || cs=='Y')
{
    int p=1,day=0;
    fflush(stdin);
    printf("\n\nEnter name:: ");
    scanf("%[^\\n]",name_s);
    printf("\n\nEnter class:: ");
    fflush(stdin);
    cls=check_class();
    printf("\n\nEnter roll:: ");
    fflush(stdin);
    scanf("%d",&rollss);
    files=fopen("student","rb+");
    temp_f=fopen("te","wb");

```



```

while(fread(&std,sizeof(std),1,files)==1)
{
    if(strcmp(std.names,name_s)==0 && clss==std.cl && rollss==std.ro)
    {
        p=0;
        printf("\n\nEnter the month till which fee to be paid:: ");
        fflush(stdin);
        scanf("%d",&monthss);
        monthss=checking_date(monthss,day);
        diffs=mm-std.dte_tme.mth;
        std.debt=(diffs*std.fes)*0.01;
        std.duedate=(diffs)*std.fes;
        if (std.debt<0) std.debt=0;
        if (std.duedate<0) std.duedate=0;
        if (std.total<0) std.total=0;
        std.total=std.debt+std.duedate+std.advanced;
        printf("\nfine :: %.2f",std.debt);
        printf("\ndue :: %.2f",std.duedate);
        printf("\ntotal :: %.2f",std.total);
        printf("\nadvance :: %.2f",std.advanced);
        std.dte_tme.mth=monthss;
        std.total=0;
        std.debt=0;
        std.duedate=0;
        fwrite(&std,sizeof(std),1,temp_f);
    }
}

if (p==1)
    printf("\n\nRECORD NOT FOUND");

```

```

printf("\n\n");
system("pause");
fflush(stdin);

fclose(files);
fclose(temp_f);
system("del student");
system("ren te, student");
printf("\n\nDo you want to continue with the process(press y or n");
fflush(stdin);
cs=getch();
}
getch();
}
void exits()
{
    int l;

    printf("\n\n\t\t Thank you for using SRM School Billing \n\n");
    system("pause");
    system("cls");
    printf("\n\n\t\t Exiting\n\n");
    for(l=1; l<=80; l++)
    {
        Sleep(50);
        printf("*");
    }
    exit(0);
}

```

```

int checking_date(int mnt,int dnt)
{
    int monthsss,daysss;
    if (mnt>12 || mnt<1 || dnt<1 || dnt>32)
    {
        MessageBox(0,"Invalid Date!\nEnter Again","Error!",0);

        fflush(stdin);

        scanf("%d%d",&monthsss,&daysss);
        monthsss=checking_date(monthsss,daysss);
    }
    else
        return (mnt);
}

int check_class()
{
    int mntttt,monsss;
    fflush(stdin);
    scanf("%d",&mntttt);
    if (mntttt>12 || mntttt<1)
    {

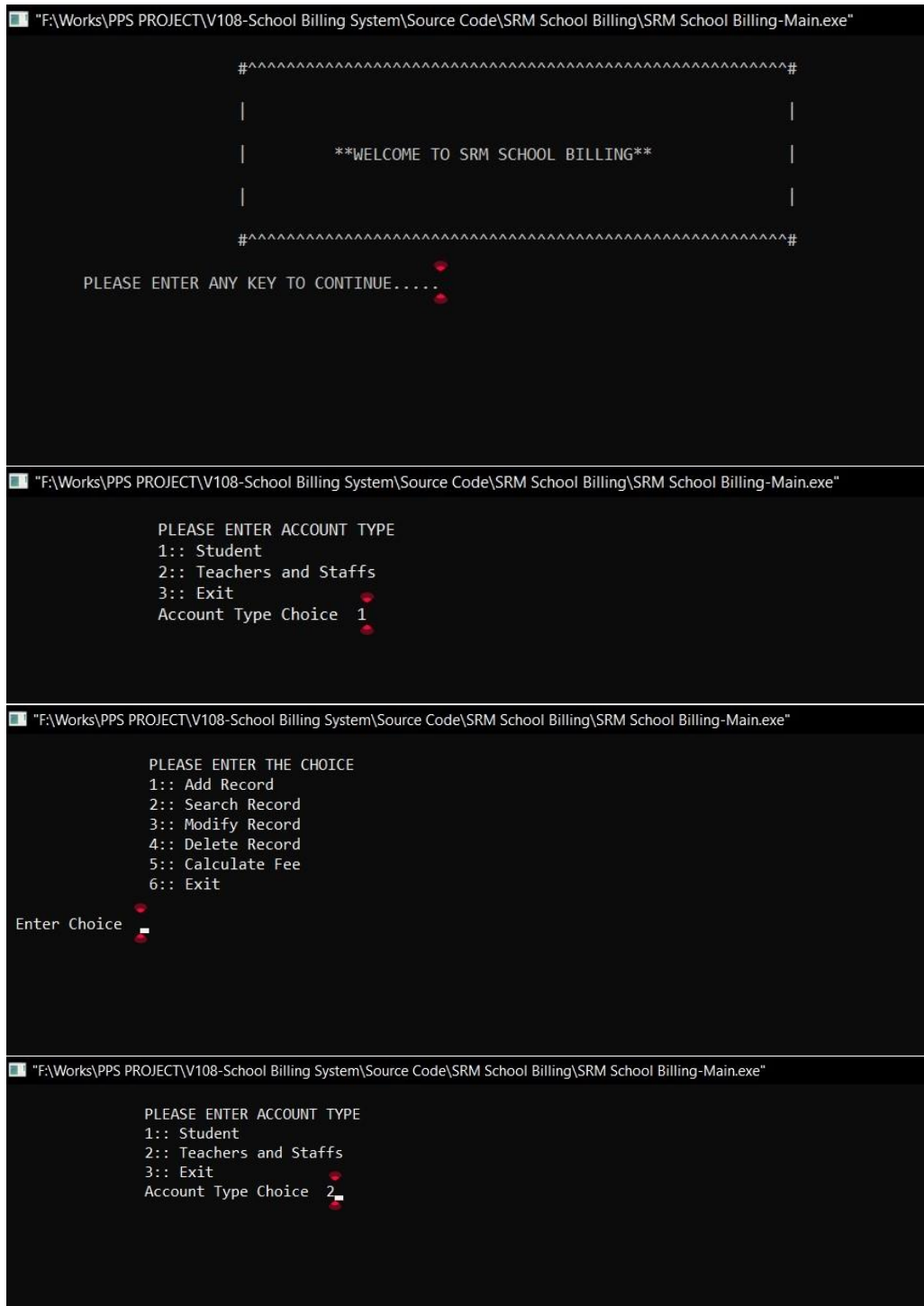
        MessageBox(0,"Invalid Class!\nEnter Class","Error!!",0);

        fflush(stdin);

        monsss=check_class();
    }
    else
        return mntttt;
}

```

SCREENSHOTS



"F:\Works\PPS PROJECT\V108-School Billing System\Source Code\SRM School Billing\SRM School Billing-Main.exe"

PLEASE ENTER THE CHOICE

- 1:: Add Record
- 2:: Search Record
- 3:: Modify Record
- 4:: Delete Record
- 5:: Calculate Salary
- 6:: Exit

Enter Choice 2

name of teacher/staff to search: Malini

name = Malini

teacher/staff id = 23

month till when salary is paid =10

monthly salary = 3000.00

advance paid = 24000.00

Press any key to continue . . .

"F:\Works\PPS PROJECT\V108-School Billing System\Source Code\SRM School Billing\SRM School Billing-Main.exe"

PLEASE ENTER ACCOUNT TYPE

- 1:: Student
- 2:: Teachers and Staffs
- 3:: Exit

Account Type Choice 3

"F:\Works\PPS PROJECT\V108-School Billing System\Source Code\SRM School Billing\SRM School Billing-Main.exe"

Exiting

RESULT

School Billing System has been executed successfully.

CONCLUSION

In conclusion, the school billing system is capable of storing school resources such as student and teacher records successfully. It helps to maintain the relation between school management and the people who have enrolled in the schools. Many people are happy with this program which helps them to store the data and records easily.