**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**

(AN AUTONOMOUS INSTITUTION)

(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, APPROVED BY AICTE & GOVT.OF KARNATAKA)

****

COURSE-PROJECT REPORT

ON

**STUDENT DATABASE MANAGEMENT**

*Submitted by:*

In partial fulfillment of the requirements for the completion of *IV Semester Object-Oriented Programming Course-Project work* during the academic year 2014-2015.

Department of Computer Science and Engineering

Nitte Meenakshi Institute of Technology,

Yelahanka, Bangalore – 560064

Academic Year 2014-15

**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**

(AN AUTONOMOUS INSTITUTION)

(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, APPROVED BY AICTE & GOVT.OF KARNATAKA)

****

**CERTIFICATE**

**This is to certify that the Project Report**

**STUDENT DATABASE MANAGEMENT**

Is an authentic work carried out by

In partial fulfillment of the requirements for the completion of *IV Semester Object-Oriented Programming Course-Project work*during the academic year 2014-2015.

Name & Signature of the Guide Name & Signature of HOD

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible,whose constant guidance and encouragement crowned our efforts with success.

We express our greatest thanks to Associate Prof. Mrs. Vijaya Shetty for her valuable efforts and support in guiding us to complete this project successfully.Without her effort it was impossible for us to finish our project.

We would also express our greatest thanks to our HOD Dr. Venkatesh K and all the faculty for their support without which this project would not have been possible.

Last but not the least, we would like to thank our family members and friends who inspired us to complete this project.

**ABSTRACT**

The program student database management uses some major concepts of OOP(object oriented programming) using C++. This program is simple and object oriented database. In this programOOP employs the bottom-up approach.

OOP treats data as a critical element in the program development and does not allow it to flow freely around the system. A class is a way to bind the data and is associated with functions together. It allows data and function to be hidden. A class is a abstract data type that can be treated like any other built-in data type. A problem is considered as a collection of number of entities called objects. Objects are instances of classes.

C++ is a statically typed, free-form, multi-paradigm, compiled, general purpose programming language. It is regarded as an intermediate-level language, as it comprises a combination of both high-level and low-level language features.

C++ is also used for hardware design, where the design is initially described in C++, then analyzed, architecturally constrained, and scheduled to create a register-transfer level hardware description language via high-level synthesis.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sl.No** | **Description** | **Page No.** |
|  | **LITERATURE SURVEY** | 6 |
|  | **INTRODUCTION** | 7 |
|  | **CORRELATING THEORETICAL CONCEPTS WITH PRACTICAL ISSUES** | 10 |
|  | **INITIAL DESIGN** | 12 |
|  | **IMPLEMENTATION** | 13 |
|  | **SNAPSHOTS** | 25 |
| 7. | **BIBLIOGRAPHY** | 29 |
| 8. | **CONCLUSION** | 30 |

**LITERATURE SURVEY**

Upon visiting and browsing about various schools and colleges, it has been found that all of them lack systematically storage of the records of student’s detail as entries were stored on a paper. As a consequence of it, it becomes tedious for a person to look through the register and obtain the details of student. We realised that a flexible system was needed to be made so that the student details marks and attendance can be maintained easily and makes the job easy.

**INTRODUCTION**

C++ is a statically typed, compiled, general-purpose, case-sensitive, free-form programming language that supports procedural, object-oriented, and generic programming.

C++ is regarded as a middle-level language, as it comprises a combination of both high-level and low-level language features.

C++ was developed by Bjarne Stroustrup starting in 1979 at Bell Labs in Murray Hill, New Jersey, as an enhancement to the C language and originally named C with Classes but later it was renamed C++ in 1983.

C++ is a superset of C, and that virtually any legal C program is a legal C++ program.

Object-Oriented Programming:

C++ fully supports object-oriented programming, including the four pillars of object-oriented development:

* Encapsulation
* Data hiding
* Inheritance
* Polymorphism

Standard Libraries:

Standard C++ consists of three important parts:

* The core language giving all the building blocks including variables, data types and literals, etc.
* The C++ Standard Library giving a rich set of functions manipulating files, strings, etc.
* The Standard Template Library (STL) giving a rich set of methods manipulating data structures, etc.

Object-Oriented Programming Concepts:

The core of the pure object-oriented programming is to create an object, in code, that has certain properties and methods. While designing C++ modules, we try to see whole world in the form of objects. For example a car is an object which has certain properties such as color, number of doors, and the like. It also has certain methods such as accelerate, brake, and so on.

There are a few principle concepts that form the foundation of object-oriented programming:

## Object:

This is the basic unit of object oriented programming. That is both data and function that operate on data are bundled as a unit called as object.

## Class:

When you define a class, you define a blueprint for an object. This doesn't actually define any data, but it does define what the class name means, that is, what an object of the class will consist of and what operations can be performed on such an object.

## Abstraction:

Data abstraction refers to, providing only essential information to the outside world and hiding their background details, i.e., to represent the needed information in program without presenting the details.

For example, a database system hides certain details of how data is stored and created and maintained. Similar way, C++ classes provides different methods to the outside world without giving internal detail about those methods and data.

## Encapsulation:

Encapsulation is placing the data and the functions that work on that data in the same place. While working with procedural languages, it is not always clear which functions work on which variables but object-oriented programming provides you framework to place the data and the relevant functions together in the same object.

## Inheritance:

One of the most useful aspects of object-oriented programming is code reusability. As the name suggests Inheritance is the process of forming a new class from an existing class that is from the existing class called as base class, new class is formed called as derived class.

This is a very important concept of object-oriented programming since this feature helps to reduce the code size.

Polymorphism:

The ability to use an operator or function in different ways in other words giving different meaning or functions to the operators or functions is called polymorphism. Poly refers to many. That is a single function or an operator functioning in many ways different upon the usage is called polymorphism.

## Overloading:

The concept of overloading is also a branch of polymorphism. When the exiting operator or function is made to operate on new data type, it is said to be overloaded.

FEATURES OF OOP:

* OOP provides a clear modular structure for programs which makes it good for defining abstract datatypes where implementation details are hidden and the unit has a clearly defined interface
* OOP makes it easy to maintain and modify existing code as new objects can be created with small differences to existing ones.
* OOP provides a good framework for code libraries where supplied software components can be easily adapted and modified by the programmer. This is particularly useful for developing graphical user interfaces.

**Correlating theoretical concepts with practical issues**

**BASIC CLASS:**

A class is a user defined data type which describes the properties and behaviours of objects.

class staff

{

Access specifier:

Data members;

Member functions;

};

**INHERITANCE:**

Objects of one class called as derived class acquires the properties and behaviors of other class called as base class.

In the following program, class admin is derived from the class staff as it acquires the properties and behaviors of class staff.

class ADMIN:public STAFF

{

…

};

**OPERATOR OVERLOADING:**

Operator overloading is a mechanism of giving a special meaning to the existing operator which behaves differently at different situations.

Operator++ is overloaded to increment the attendance of a student. Potfix++ operator is used.

//argument int is used as a dummy parameter

void operator++(int){….}

**INLINE FUNCTION:**

An inline function is one for which the compiler copies the code from the function definition directly into the code of the calling function rather than creating a separate set of instructions in memory.

We have used two inline functions in the following program:

* clrscr() which is used to clear the screen
* title() which displays the title

**VIRTUAL FUNCTIONS:**

It is a member function of base class which is redefined in the derived class.

manageAttendance() is used as virtual function in staff class and overridden in admin class as it behaves differently for both objects.

Class STAFF { //members

Virtual int manageAttendence(){ // body of the function

}

Class ADMIN{

//members of the class

int manageAttendence(){ // body of the function

}

};

**FILES:**

Files are used to maintain database.

In the following program, student.dat file contains the detail of student. Staff.dat file contains the detail of staff. Other files like log.bin math.dat oop.dat co.dat etc. are used as temporary files.

**INITIAL DESIGN**

|  |
| --- |
| **1 : STUDENT**  **2 : STAFF**  **3 : ADMIN**  **4 : GUEST** |

|  |  |  |  |
| --- | --- | --- | --- |
| **STUDENT**  1.View Profile  2.View Attendance  3.View Marks  4.View Notifications  5.Give a Complaint  6.Upload Papers  7.View Exam Timetable | **STAFF**  1.View Profile  2.Manage attendance  3.Manage Marks | **ADMIN**  1.Add Student  2.Remove student  3.Add staff  4.Remove staff  5.List Students  6.Manage Attendance  7.View Student’s Attendance  8.View student’s marks  9.Add notification  10.Check Complaints  11.Upload Exam Timetable | **GUEST**  Can view all the papers uploaded by the students |

**IMPLEMENTATION**

#include<stdlib.h>

#include<conio.h>

#include<cstring>

#include<iostream>

#include<fstream>// FILE HANDLING

#include<windows.h>

#include<time.h>

#include<cstdio>

#define F\_NOT\_FOUND -1

#define USN\_NOT\_FOUND -2

#define SUCCESS 0

#define ADMIN\_PASS "OOP"

using namespace std;

int flag,subject=0;

inline void clrscr(){

system("cls");

}

inline void title(){

clrscr();

cout<<"==============================================================================="<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Institute Student Database Management System \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"==============================================================================="<<endl;

}

void getdate(char \*tday){

time\_t tim;

tim=time(NULL);

struct tm \*d=localtime(&tim);

//char tday[10];

sprintf(tday,"%02d/%02d/%d",d->tm\_mday,d->tm\_mon+1,d->tm\_year+1900);

}

//sets current date in file

void setbits(){

ofstream setb;

char today[11];

getdate(today);

switch(subject){

case 1 : setb.open("math.dat");

setb<<today;

break;

case 2 :

setb.open("co.dat");

setb<<today;

break;

case 3 :

setb.open("mp.dat");

setb<<today;

break;

case 4 :

setb.open("oop.dat");

setb<<today;

break;

}

setb.close();

};

//return true if attendence is already entered

bool isAt\_Done(){

char entered[11],today[11];

ifstream setb;

switch(subject){

case 1 :

setb.open("math.dat");

break;

case 2 :

setb.open("co.dat");

break;

case 3 :

setb.open("mp.dat");

break;

case 4 :

setb.open("oop.dat");

break;

}

setb.getline(entered,15);

setb.close();

getdate(today);

if(!strncmp(entered,today,10)) return true;

else return false;

}

class STUDENT; //forward declaration of class STUDENT;

class STAFF {

protected :

char staff\_id[10];

char staff\_name[40];

char staff\_sub[5];

char email[30];

public : STAFF(){

staff\_id[0]=staff\_name[0]=staff\_sub[0]=email[0]='\0';

}

STAFF(char \*id,char \*name,char \*sub,char \*mail){

strcpy(staff\_id,id);

strcpy(staff\_name,name);

strcpy(staff\_sub,sub);

strcpy(email,mail);

}

char\* getid(){

return staff\_id;

}

int init(char\*);

int manageAttendence();

int manageMarks();

virtual void showProfile(){

clrscr();

title();

cout<<"My proifile...."<<endl;

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

cout<<" Staff Name \t :"<<staff\_name<<endl;

cout<<" Staff Id \t :"<<staff\_id<<endl;

cout<<" Department \t :"; if(subject==1) cout<<"Mathematics"<<endl;

else cout<<"Computer Science And Engineering"<<endl;

cout<<" Email id \t :"<<email<<endl;

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

fflush(stdin);

cin.get();

}

};

// admin class inherited from <class>STAFF for managing student data with administrative account

class ADMIN:public STAFF{

//admin attributes are similar to staff, so same members of staff is used here for admin data.

char pass[10];

public : ADMIN(char id[]){

strcpy(staff\_id,id);

strcat(id,".dat");

ifstream fp(id,ios::in);

if(fp){

fp>>pass;

//pass = ADMIN\_PASS;

fp>>staff\_name;

fp>>email;

}

}

void setsub(char \*s){ strcpy(staff\_sub,s); }

int addstaff();

int admitStudent();

int removeStaff(char\*);

int removeStudent(char\*);

void viewMarks();

void viewAttendence();

int manageAttendence();

int listStudent();

int Notification();

int complaint();

int timetable();

bool isValidPass(char \*p){

cout<<"\n\n\tAuthenticating..."<<p<<" = "<<ADMIN\_PASS;

Sleep(500);// has to be removed or else password "CSE" ll be visible

if(!strcmp(ADMIN\_PASS,p)){

return true;

}else {false;

}

}

void showProfile();

};

class STUDENT{

int admissionNo;

char usn[11];

char name[30];

char email[30];

char dob[11];

struct sub{

int mp,co,oop,math;

}marks,attendence;

public : STUDENT(){

admissionNo=-1;

strcpy(usn,"\0");

strcpy(usn,"\0");

strcpy(email,"\0");

strcpy(dob,"\0");

marks.mp=-1;

marks.co=-1;

marks.oop=-1;

marks.math=-1;

attendence.mp=0;

attendence.co=0;

attendence.oop=0;

attendence.math=0;

}

//Member fucntion prototype

int init(char\*);

void showProfile();

void showAttendence();

void showMarks();

int Notification();

int complaint();

int paper();

int timetable();

friend class ADMIN;

friend class STAFF;// imp

void operator++(int);//operator function

};

void ADMIN::showProfile(){

clrscr();

title();

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

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

cout<<" Admin Name \t :"<<staff\_name<<endl;

cout<<" Admin Id \t :"<<staff\_id<<endl;

cout<<" Email Id \t :"<<email<<endl;

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

fflush(stdin);

cin.get();

}

void ADMIN:: viewMarks(){

char usn[15];

clrscr();

title();

cout<<endl<<"Please enter the student USN : "; cin>>usn;

strupr(usn);

STUDENT s;

fflush(stdin);

if(s.init(usn)){

cout<<" Enter valid USN...";

cin.get();

return;

}

s.showMarks();

}

void ADMIN::viewAttendence(){

char u[15];

clrscr();

title();

cout<<"Please enter student USN : "; cin>>u;

strupr(u);

STUDENT s;

fflush(stdin);

if(s.init(u)){

cout<<" Enter valid USN...";

cin.get();

return;

}

s.showAttendence();

}

// remove student ::by setting up all the attributes of student to null and overwriting on the old blocks

int ADMIN::removeStudent(char \*U){

bool flag=false;

long pos;

fstream readfile("student.dat",ios::binary|ios::in|ios::out);

if(!readfile){

cerr<<"Error in opening the file...please try again";

cin.get();

return F\_NOT\_FOUND;

}

STUDENT temp;

while(!readfile.eof()){

pos=readfile.tellg(); //replace object with null bytes

readfile.read((char\*)&temp,sizeof(temp));

if(!strcmp(U,temp.usn)){

flag=true;

break;

}

}

if(!flag){

cerr<<"\tInvalid user "<<endl;

return USN\_NOT\_FOUND;

cin.get();

}

//else initialize student fields

temp.admissionNo=0;

strcpy(temp.name,"\0");

strcpy(temp.usn,"\0");

strcpy(temp.email,"\0");

strcpy(temp.dob,"\0");

temp.marks.mp=0;

temp.marks.co=0;

temp.marks.oop=0;

temp.marks.math=0;

temp.attendence.mp=0;

temp.attendence.co=0;

temp.attendence.oop=0;

temp.attendence.math=0;

readfile.seekp(pos);

readfile.write((char\*)&temp,sizeof(temp));

fflush(stdin);

cout<<"Student record removed successfully...";

cin.get();

return SUCCESS;

}

int ADMIN::addstaff(){

char name[40],sub[5],email[30],id[10];

clrscr();

ofstream putf("staff.dat",ios::binary|ios::out|ios::app);

if(!putf){

cerr<<"Unable to open student record file ";

return F\_NOT\_FOUND;

}

title();

cout<<" Please enter the following fields...[all fields are mandatory]"<<endl;

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

cout<<"\n Enter staff id : "; cin>>id;

fflush(stdin);

cout<<"\n Enter staff name : "; cin.getline(name,30);

fflush(stdin);

cout<<"\n Enter subject to handle[co/oop/mp/math] : "; cin>>sub;

strlwr(sub);

fflush(stdin);

cout<<"\n Enter the mail id : "; cin>>email;

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

STAFF temp(id,name,sub,email);

fflush(stdin);

cout<<"Press enter to proceed "; cin.get();

clrscr();

temp.showProfile();

putf.write((char\*)&temp,sizeof(temp));

putf.close();

cout<<endl<<" Staff member added successfully..."; cin.get();

return SUCCESS;

}

int ADMIN::Notification(){

clrscr();

ofstream putf("notification.dat",ios::binary|ios::out|ios::app);

if(!putf){

cerr<<"Unable to open student record file ";

return F\_NOT\_FOUND;

}

title();

char today[11],note[500];

getdate(today);

cout<<"\t\t\t\t\t\t\t Today : "<<today;

cout<<"\nEnter Notifications: \n";

fflush(stdin);

cin.getline(note,500);

putf<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ "<<today<<"\n"<<note<<"\n\n\n";

putf.close();

cout<<endl<<" Notification added successfully..."; cin.get();

}

int ADMIN::complaint(){

clrscr();

char ch;

ifstream readfile1("complaint.dat",ios::binary|ios::in);

if(!readfile1.is\_open()){

return F\_NOT\_FOUND;

}

fflush(stdin);

title();

readfile1.seekg(0);

cout<<"\n ";

while(!readfile1.eof()){

readfile1.get(ch);

cout<<ch;

}

cout<<"\nPress any key to return...";

cin.get();

readfile1.close();

}

int ADMIN::timetable(){

clrscr();

ofstream pf("table.dat",ios::binary|ios::out);

if(!pf){

cerr<<"Unable to open student record file ";

return F\_NOT\_FOUND;

}

title();

char today[11],note[20],start[10],end[10];

getdate(today);

cout<<"\t\t\t\t\t\t\t Today : "<<today;

cout<<"\nEnter exam starting and ending time: \n";

cin>>start>>end;

fflush(stdin);

pf<<"\n\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ "<<today<<"\n\n\n";

cout<<"Enter exam date for following subject :\n";

pf<<'|'<<" SUBJECT "<<'|'<<"\t"<<"DATE "<<'|'<<"\t"<<"START TIME "<<'|'<<"\t"<<"END TIME "<<" "<<'|'<<endl;

cout<<"OOP : ";

cin>>note;

pf<<'|'<<" OOP "<<'|'<<"\t"<<note<<" "<<'|'<<"\t"<<start<<" "<<'|'<<"\t"<<end<<" "<<'|'<<endl;

cout<<"MATHS : ";

cin>>note;

pf<<'|'<<" MATHS "<<'|'<<"\t"<<note<<" "<<'|'<<"\t"<<start<<" "<<'|'<<"\t"<<end<<" "<<'|'<<endl;

cout<<"CO : ";

cin>>note;

pf<<'|'<<" CO "<<'|'<<"\t"<<note<<" "<<'|'<<"\t"<<start<<" "<<'|'<<"\t"<<end<<" "<<'|'<<endl;

cout<<"MP : ";

cin>>note;

pf<<'|'<<" MP "<<'|'<<"\t"<<note<<" "<<'|'<<"\t"<<start<<" "<<'|'<<"\t"<<end<<" "<<'|'<<endl;

pf.close();

cout<<endl<<" Exam time table updated successfully..."; cin.get();

cin.get();

}

int ADMIN::removeStaff(char \*U){

bool flag=false;

long pos;

fstream readfile("staff.dat",ios::binary|ios::in|ios::out);

if(!readfile){

cerr<<"Error in opening file...please try again";

return F\_NOT\_FOUND;

}

STAFF temp;

while(!readfile.eof()){

pos=readfile.tellg(); //replace object with null bytes

readfile.read((char\*)&temp,sizeof(temp));

if(!strcmp(U,temp.getid())){

flag=true;

break;

}

}

if(!flag){

cerr<<"\tInvalid user "<<endl;

return USN\_NOT\_FOUND;

}

readfile.seekp(pos);

STAFF del;

readfile.write((char\*)&del,sizeof(STAFF));

fflush(stdin);

cout<<"Staff removed successfully...";

cin.get();

return SUCCESS;

}

//administrator acces to change the attendenc

int ADMIN::manageAttendence(){

title();

char tday[10],date[11];

flag=0;

getdate(tday);

cout<<"\t\t\t\t\t\t\t Today : "<<tday;

cout<<"\n\tYou have permission to change the attendence, only 6 days before the"<<endl;

cout<<" final exam to give eligibilty to the student who has less than 75%."<<endl;

cout<<endl<<"\n\nEnter the final exam date [DD/MM/YYYY] : ";

cin>>date;

int i,j,x,y;

x=tday[3]-'0';

x=x\*10+(tday[4]-'0');

y=date[3]-'0';

y=y\*10+(date[4]-'0');

i=tday[0]-'0';

i=i\*10+(tday[1]-'0');

j=date[0]-'0';

j=j\*10+(date[1]-'0');

Sleep(1000);

if(j-i<7&&x==y){

cout<<"Enter the subject to modify :";

cin>>staff\_sub;

fflush(stdin);

strlwr(staff\_sub);

cout<<"Enter the USN : ";

fflush(stdin);

char u[11];

cin>>u;

strupr(u);

STUDENT s;

fstream f("student.dat",ios::in|ios::in|ios::out);

int pos;

while(!f.eof()){

pos=f.tellg();

f.read((char\*)&s,sizeof(s));

if(!strcmp(s.usn,u)){

flag=1;

break;

}

}

if(flag){

cout<<"Enter the attendence : ";

int num;

cin>>num;

if(subject==3)

s.attendence.mp=num;

else if(subject==2)

s.attendence.co=num;

else if(subject==4)

s.attendence.oop=num;

else if(subject==1)

s.attendence.math=num;

f.seekp(pos);

f.write((char\*)&s,sizeof(s));

cout<<"Attendence updated.";

f.close();

fflush(stdin);

cin.get();

}

else {cerr<<"USN not found...";

cin.get();

}

}

else {

cout<<"Sorry, You cannot change the attendance.";

Sleep(800);

return 0;

}

}

int ADMIN::listStudent(){

title();

char prev[12];

ifstream readF("student.dat",ios::binary|ios::in);

STUDENT temp;

fflush(stdin);

cout<<"\tUSN\t\tNAME"<<endl;

int full=0;

while(!readF.eof()){

readF.read((char\*)&temp,sizeof(temp));

if(temp.admissionNo==-1){ cout<<endl<<"End of list";

break;

}

full++;

if(full>10){

cout<<"Press enter to continue...";

cin.get();

full=0;

title();

cout<<"\tUSN\t\tNAME"<<endl;

}

if(!strcmp(prev,temp.usn)) continue;

strcpy(prev,temp.usn);

cout<<"\t"<<temp.usn<<"\t"<<temp.name<<endl;

}

cout<<"\nPress any key to return...";

cin.get();

readF.close();

}

// admit student

// Administrator should provide all the fields required.

int ADMIN::admitStudent(){

clrscr();

fstream putf("student.dat",ios::binary|ios::out|ios::app);

if(!putf){

cerr<<"Unable to open student record file ";

return F\_NOT\_FOUND;

}

STUDENT temp;

putf.seekp(ios::end);

title();

cout<<"Please enter the following fields...[all fields are mandatory]"<<endl;

fflush(stdin);

STUDENT t;

int status=0;

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

cout<<"\n # Please enter usn number : "; cin>>temp.usn;

fflush(stdin);

strupr(temp.usn);

if(USN\_NOT\_FOUND!=t.init(temp.usn)){

cout<<"This USN number is already assigned to : "<<t.name;

Sleep(1000);

if(++status<3)

admitStudent();

else return 0;

}

fflush(stdin);

cout<<"\n # Please enter full name of the student : "; cin.getline(temp.name,30);

fflush(stdin);

cout<<"\n # Please enter date of birth : "; cin>>temp.dob;

fflush(stdin);

cout<<"\n # Please enter new admission number : "; cin>>temp.admissionNo;

fflush(stdin);

fflush(stdin);

cout<<"\n # Please enter email id of the student : "; cin>>temp.email;

fflush(stdin);

cout<<"\n Press enter to proceed ";

cin.get();

clrscr();

//genusn(temp.usn);

//strcpy(temp.usn,"1NT13CS065");

//temp.admissionNo=12339;

title();

cout<<"\n The new student details are as given below... "<<endl;

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

cout<<"\n # Admission Number : "<<temp.admissionNo<<endl;

cout<<" # Student name : "<<temp.name<<endl;

cout<<" # University Seat Number : "<<temp.usn<<endl;

cout<<" # Date of birth : "<<temp.dob<<endl;

cout<<" # Email id : "<<temp.email<<endl;

temp.attendence.mp=0;

temp.attendence.co=0;

temp.attendence.oop=0;

temp.attendence.math=0;

cout<<" Are the above details correct? [Y/N] : ";

char yes;

cin.get(yes); fflush(stdin);

if(yes=='Y'||yes=='y') {

putf.write((char\*)&temp,sizeof(temp));

cout<<"New student added successfully..."<<endl;

cin.get();

putf.close();

}

else {

cout<<" Student details weren't added ";

Sleep(800);

return -1;

}

clrscr();

}

void STUDENT::showProfile(){

clrscr();

title();

cout<<"My proifile...."<<endl;

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

cout<<" Name :"<<name<<endl;

cout<<" Admission number :"<<admissionNo<<endl;

cout<<" USN :"<<usn<<endl;

cout<<" Branch : Computer Science And Engineering"<<endl;

cout<<" Date of birth :"<<dob<<endl;

cout<<" Email id :"<<email<<endl;

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

cout<<"Press any key to exit :";

cin.get();

}

//display marks

void STUDENT::showMarks(){

clrscr();

title();

cout<<" Test Result : "<<name<<endl;

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

cout<<" SUBJECT NAME | MARKS "<<endl;

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

cout<<" Micro Processor | "; (marks.mp>=0)?cout<<marks.mp<<endl:cout<<"Not Entered"<<endl;

cout<<" Object Oriented Programming | "; (marks.oop>=0)?cout<<marks.oop<<endl:cout<<"Not Entered"<<endl;

cout<<" Computer Organization & architector | "; (marks.co>=0)?cout<<marks.co<<endl:cout<<"Not Entered"<<endl;

cout<<" Mathematics | "; (marks.math>=0)?cout<<marks.math<<endl:cout<<"Not Entered"<<endl;

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

cout<<"Press any key to exit :";

cin.get();

}

//dispaly attendencce

void STUDENT::showAttendence(){

fflush(stdin);

clrscr();

title();

cout<<" Attendence Result : "<<name<<endl;

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

cout<<" SUBJECT NAME | Class Attended "<<endl;

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

cout<<" Micro Processor | "<<attendence.mp<<endl;

cout<<" Object Oriented Programming | "<<attendence.oop<<endl;

cout<<" Computer Organization & architector | "<<attendence.co<<endl;

cout<<" Mathematics | "<<attendence.math<<endl;

cout<<"Press any key to exit :"; cin.get();

}

//initialize the student data with details of logged in student

// also checks if log-in id(USN) is correct/exist

int STUDENT::init(char \*U){

bool flag=false;

ifstream readfile("student.dat",ios::binary|ios::in);

if(!readfile.is\_open()){

return F\_NOT\_FOUND;

}

STUDENT temp;

readfile.seekg(0);

while(!readfile.eof()){

readfile.read((char\*)&temp,sizeof(temp));

if(!strcmp(U,temp.usn)){

flag=true;

break;

}

if(temp.admissionNo==-1)

break;

}

if(!flag){

return USN\_NOT\_FOUND;

}

//else initialize student fields

\*this=temp;

return SUCCESS;

}

int STUDENT::Notification(){

clrscr();

char ch;

ifstream readfile("notification.dat",ios::binary|ios::in);

if(!readfile.is\_open()){

return F\_NOT\_FOUND;

}

title();

readfile.seekg(0);

cout<<"\n ";

while(!readfile.eof()){

readfile.get(ch);

cout<<ch;

}

cout<<"\nPress any key to return...";

cin.get();

readfile.close();

}

int STUDENT::timetable(){

clrscr();

char ch;

ifstream readfile1("table.dat",ios::binary|ios::in);

if(!readfile1.is\_open()){

return F\_NOT\_FOUND;

}

title();

cout<<"\t\t\tExam Time Table\n";

readfile1.seekg(0);

cout<<"\n ";

while(!readfile1.eof()){

readfile1.get(ch);

cout<<ch;

}

cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ";

cout<<"\nPress any key to return...";

cin.get();

readfile1.close();

}

char u1[11];

int STUDENT::complaint(){

clrscr();

ofstream putf("complaint.dat",ios::binary|ios::out|ios::app);

if(!putf){

cerr<<"Unable to open student record file ";

return F\_NOT\_FOUND;

}

title();

char today[11],note[500];

getdate(today);

cout<<"\t\t\t\t\t\t\t Today : "<<today;

cout<<"\nEnter Complaint: \n";

fflush(stdin);

cin.getline(note,500);

putf<<"\n\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ "<<today<<"\n"<<note<<"\t -"<<u1<<"\n\n\n";

putf.close();

cout<<endl<<" Complaint given successfully..."; cin.get();

}

int STUDENT::paper(){

clrscr();

char today[11],note[1000],ch;

getdate(today);

ofstream f;

char filename[40];

cout << "Please enter your paper name : ";

cin.getline(filename,40);

int length = strlen(filename);

int x = 0;

while (x != length) {

ch=filename[x];

if (ch==' ')

filename[x] ='\_';

x++;}

ofstream pf("topic.dat",ios::binary|ios::out|ios::app);

pf<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ "<<today<<"\n"<<filename<<"\t -"<<u1<<"\n";

strcat(filename,".dat");

f.open( filename,ios::binary|ios::out);

// ofstream putf("complaint.dat",ios::binary|ios::out|ios::app);

if(!f){

cerr<<"Unable to open file ";

return F\_NOT\_FOUND;

}

title();

getdate(today);

cout<<"\t\t\t\t\t\t\t Today : "<<today;

cout<<"\nType your paper: \n";

fflush(stdin);

cin.getline(note,1000);

f<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ "<<today<<"\n"<<note<<"\t -"<<u1<<"\n\n\n";

pf.close();

f.close();

cout<<endl<<" Paper Uploaded successfully..."; cin.get();

}

int STAFF::init(char \*id){

bool flag=false;

ifstream readfile("staff.dat",ios::binary|ios::in);

if(!readfile.is\_open()){

cerr<<"\nError opening the record... please try again."<<endl;

Sleep(3000);

return F\_NOT\_FOUND;

}

STAFF temp;

while(!readfile.eof()){

readfile.read((char\*)&temp,sizeof(temp));

if(!strcmp(id,temp.staff\_id)){

flag=true;

break;

}

}

if(!flag){

cout<<"\nStaff with staff id : ["<<id<<"] not found. Enter valid id"<<endl;

Sleep(1000);

return USN\_NOT\_FOUND;

}

\*this=temp;

if(!strcmp(staff\_sub,"co")) subject=2;

else if(!strcmp(staff\_sub,"mp")) subject=3;

else if(!strcmp(staff\_sub,"oop")) subject=4;

else if(!strcmp(staff\_sub,"math")) subject=1;

return SUCCESS;

}

int STAFF::manageMarks(){

title();

int c=0;

int m;

STUDENT temp;

fstream readfile("student.dat",ios::binary|ios::in|ios::out);

if(!readfile.is\_open()){

cout<<"Error : Student database file missing...";

return F\_NOT\_FOUND;

}

cout<<"Enter the max. marks and press enter key : "; cin.get();

clrscr();

title();

cout<<endl<<"Subject : ";

switch(subject){

case 1 :cout<<"Mathematics"; break;

case 2 :cout<<"Computer Organization"; break;

case 3 :cout<<"Micro Processor"; break;

case 4 :cout<<"Object Oriented Programming"; break;

}

cout<<endl;

readfile.seekg(0L);

while(!readfile.eof()){

fflush(stdin);

readfile.read((char\*)&temp,sizeof(temp));

if(!strcmp(staff\_sub,"mp")) {

cout<<"#"<<temp.usn<<" ["<<temp.marks.mp<<"]"<<"=>";

cin>>m;

temp.marks.mp=m;

}

else if(!strcmp(staff\_sub,"co")) {

cout<<"#"<<temp.usn<<" ["<<temp.marks.co<<"]"<<"=>";

cin>>m;

temp.marks.co=m;

}

else if(!strcmp(staff\_sub,"oop")) {

cout<<"#"<<temp.usn<<" ["<<temp.marks.oop<<"]"<<"=>";

cin>>m;

temp.marks.oop=m;

}

else if(!strcmp(staff\_sub,"math")) {

cout<<"#"<<temp.usn<<" ["<<temp.marks.math<<"]"<<"=>";

cin>>m;

temp.marks.math=m;

}

//write back the changes amde to the attendence.

readfile.seekp((long)readfile.tellg()-sizeof(temp));

readfile.write((char\*)&temp,sizeof(temp));

c++;

if(c>15) {

clrscr();

title();

c=0;

}

fflush(stdin);

cin.get();

}

cout<<" Marks have been updated succesfully..."; Sleep(3000);

return SUCCESS;

}

int STAFF::manageAttendence(){

char day[10];

clrscr();

fflush(stdin);

title();

getdate(day);

if(isAt\_Done()){

cout<<"\tDate : "<<day<<endl;;

cout<<"You have already entered today's attendence...\nSorry, You cannot enter again..."<<endl;

cin.get();

return 0;

}

char presence;

STUDENT temp[80];

ifstream readfile("student.dat",ios::binary|ios::in);

if(!readfile.is\_open()){

cout<<"Error : Student database file missing...";

return F\_NOT\_FOUND;

}

int n=0;

while(!readfile.eof()){

readfile.read((char\*)&temp[n],sizeof(temp[n]));

if(!strcmp(temp[n].usn,"\0")) continue;

n++;

}

readfile.close();

cout<<endl<<"Subject : ";

switch(subject){

case 1 :cout<<"Mathematics"; break;

case 2 :cout<<"Computer Organization"; break;

case 3 :cout<<"Micro Processor"; break;

case 4 :cout<<"Object Oriented Programming"; break;

}

cout<<"\tDate : "<<day<<endl;;

cout<<"\t 1 : Mark all absent\n\t 2 : Enter absenties' usn "<<endl;

cout<<"\t Select your choice : ";

int choice;

char yes;

cin>>choice;

if(choice==1){ cout<<"Marking everyone absent...\nAre you sure : [Y/N] ";

cin>>yes;

if(yes=='y'||yes=='Y'){

setbits();

cout<<"Attendence updated...";

return 0;

}

}

else if(choice!=2) return 0;

clrscr();

title();

cout<<endl<<"Enter absenties usn : [Complete USN or last 3/2 digit(0 to end) ]\n\_";

char absent[12],l\_absent[12]; //ASSUMED ONLY FOR CSE DEPT

int count=0;

fstream t("log.bin",ios::in|ios::out|ios::trunc);

while(1){

cin>>absent;

if(!strcmp(absent,"0")) break;

strupr(absent);

if((strlen(absent))==3){

strcpy(l\_absent,"1NT13CS");

strcat(l\_absent,absent);

t<<l\_absent<<endl;

}

else if((strlen(absent))==2){

strcpy(l\_absent,"1NT13CS0");

strcat(l\_absent,absent);

t<<l\_absent<<endl;

}

else

t<<absent<<endl;

count++;

}

t<<"0";

title();

t.seekp(0L);

cout<<" Absenties list : \n"<<endl;

while(1){

t.getline(absent,11);

if(!strcmp(absent,"0")) break;

cout<<" \t"<<absent<<endl;

}

t.close();

fflush(stdin);

cout<<"\n\tTotal absent = "<<count<<endl;

cout<<"\tTotal present = "<<n-count<<endl;

cout<<"Press enter to proceed : ";

cin.get();

ofstream writefile("student.dat",ios::binary|ios::out);

for(int i=0;i<n;i++){

temp[i]++;

writefile.write((char\*)&temp[i],sizeof(temp[i]));

}

setbits();

cout<<"Attendence updated...";

cin.get();

writefile.close();

}

//Upadte all student attendence, except absenties

void STUDENT::operator++(int){

flag=0;

ifstream log("log.bin");

char absent[11];

log.seekg(0L);

while(!log.eof()){

log.getline(absent,11);

if(!strncmp(usn,absent,10)){

flag=1; break;

}

}

if(flag==0){

if(subject==3)

++attendence.mp;

else if(subject==2)

attendence.co++;

else if(subject==4)

attendence.oop++;

else if(subject==1)

attendence.math++;

} log.close();

}

//Initial usage...Program will check for the availabe student database files

//if its not available program will create new files

int setup(){

title();

ifstream fp("student.dat",ios::binary|ios::in);

if(fp.is\_open())

return 0;

cout<<endl<<"Initiating required files for first time usage... ";

fp.close();

ofstream ofp("student.dat",ios::binary|ios::out);

// Store the initial data to the file.

if(!ofp.is\_open()) return -1;

ofp.close();

ofstream \_ofp("staff.dat",ios::binary|ios::out);

if(!\_ofp.is\_open()) return -1;

\_ofp.close();

ofstream \_o("log.bin",ios::out);

\_o.close();

ofstream \_f("notification.dat",ios::binary|ios::out);

if(!\_f.is\_open()) return -1;

\_f.close();

ofstream \_fi("complaint.dat",ios::binary|ios::out);

if(!\_fi.is\_open()) return -1;

\_fi.close();

ofstream \_fii("topic.dat",ios::binary|ios::out);

if(!\_fii.is\_open()) return -1;

\_fii.close();

Sleep(2000);

cout<<" : DONE";

cin.get();

return 0;

}

// User handler function, prototype;

int student(),staff(),guest(), admin();

int main(){

if(setup()) {

cout<<"Files are missing and unable to create new files...\n Please try again..."; Sleep(2000);

exit(1);

}

while(1){

clrscr();//inline fn

title();//inline fn

cout<<"\n\t\tLogin Option"<<endl;

cout<<"\t\t 1 : Student "<<endl;

cout<<"\t\t 2 : Staff "<<endl;

cout<<"\t\t 3 : Admin "<<endl;

cout<<"\t\t 4 : Guest "<<endl;

cout<<"\t\t 5 : Exit"<<endl;

cout<<endl<<endl<<"\t\tEnter your choice : ";

static int error;

int logChoice;

cin>>logChoice;

switch(logChoice) {

case 1 : error=0;

while(error<4) { //each time student gets max of 4 attempts to enter correct password or id

if(student())

error++;

else break;

}

if(error>3)

cerr<<" Authentication failed...";//console error: std o/p stream for errors

cin.get();// similar to getch()

break;

case 2 : error=0;

while(error<4) { if(staff())

error++;

else break;

}

if(error>3) cerr<<" Authentication failed..."; cin.get();

break;

case 3 : admin();

break;

case 4 :guest();

break;

case 5 : cout<<"Terminating..."; Sleep(1000); //from WINAPI Sleep(DWORD)

exit(0);

default : cout <<"Invalid choice : ";

Sleep(1000);

cin.get();

}

}

return 0;

}

int student(){

static int error;

char usn[11];

clrscr();

title();

cout<<"\tLogin type : Student."<<endl;

cout<<endl<<"\tEnter your usn number : ";

cin>>usn;

strupr(usn);

cout<<endl<<"\tEnter the password : ";

char pass[15],c;

short i=0;

while((c=getch())!=13){ pass[i]=c;

i++; cout<<'\*';}

pass[i]='\0';

strupr(pass);

if(strcmp(usn,pass)){

cout<<"\n\tInavlid usn or password..."<<endl;

return -1;

}

int choice;

STUDENT s;

flag=s.init(usn);

if(USN\_NOT\_FOUND==flag){

cout<<"\n\tNo such usn assigned to student... ";

cin.get();

return 0;

}

else if(flag==F\_NOT\_FOUND){ cerr<<"\n\tError in opening file...please try again";

Sleep(2000);

return 0;

}

//object of student class initialized with usn

while(1) {

clrscr();

title();

cout<<"\tLogin type : Student ["<<usn<<"]"<<endl;

cout<<"\t\tMenu"<<endl;

cout<<"\t\t1 : View profile\n\t\t2 : View attendence\n\t\t3 : View marks\n\t\t4 : View Notifications \n\t\t5 : Give a Complaint \n\t\t6 : Upload papaers \n\t\t7 : Exam Time Table \n\t\t8 : Exit"<<endl;

cout<<"\tEnter your choice : ";

cin>>choice;

switch(choice){

case 1 : fflush(stdin);

clrscr();

s.showProfile();

break;

case 2 : fflush(stdin);

clrscr();

s.showAttendence();

break;

case 3 : fflush(stdin);

clrscr();

s.showMarks();

break;

case 4 : fflush(stdin);

clrscr();

s.Notification();

break;

case 5 : fflush(stdin);

clrscr();

strcpy(u1,usn);

s.complaint();

break;

case 6 : fflush(stdin);

clrscr();

strcpy(u1,usn);

s.paper();

break;

case 7 : fflush(stdin);

clrscr();

s.timetable();

break;

case 8 : cout<<"Logging out...["<<usn<<"]";

Sleep(1000);

return SUCCESS;

default : cout<<"Select valid choice : ";

Sleep(1000);

}

}

}

int staff(){

char staffid[10];

title();

cout<<"\tLogin type : Staff"<<endl;

fflush(stdin);

cout<<endl<<"\tEnter staff id : ";

cin>>staffid;

fflush(stdin);

cout<<"\tEnter the password : ";

char pass[15],c;

short i=0;

while((c=getch())!=13){pass[i]=c; i++; cout<<'\*';

}

pass[i]='\0';

strupr(pass);

strupr(staffid);

if(strcmp(staffid,pass)){

cout<<"Inavlid id/password..."<<endl;

return -1;

}

STAFF staf;

if(USN\_NOT\_FOUND==(flag=(staf.init(staffid)))){

cin.get();

return 0;

}

else if(flag) return 0;

//object of student class initialized with usn

while(1){

clrscr();

title();

cout<<"\tLogin type : Staff ["<<staffid<<"]"<<endl;

cout<<"\t\tMenu"<<endl;

cout<<"\t\t1 : View profile\n\t\t2 : Manage attendence\n\t\t3 : Manage marks\n\t\t4 : Exit"<<endl;

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

int choice;

cin>>choice;

switch(choice){

case 1 :fflush(stdin); clrscr();

staf.showProfile();

break;

case 2 : fflush(stdin); clrscr();

staf.manageAttendence();

break;

case 3 : fflush(stdin); clrscr();

staf.manageMarks();

break;

case 4 : cout<<"Logging out...["<<staffid<<"]";

Sleep(1000);

return SUCCESS;

default : cout<<"Select valid choice : ";

Sleep(1000);

}

}

}

int admin(){

char id[10];

char pass[15],c;

clrscr();

title();

cout<<"\tLogin type : Administrator"<<endl;

cout<<"\n\tEnter admin id : ";

cin>>id;

fflush(stdin);

strupr(id);

strcpy(pass,id);

ADMIN ad(pass);

cout<<"\n\tEnter admin password : ";

short i=0;

while((c=getch())!=13){ pass[i]=c; i++; cout<<'\*';

}

pass[i]='\0';

if(!ad.isValidPass(pass)){

cout<<"\r\tAuthentication error..."<<pass<<endl;

cin.get();

exit(0);

}

//object of student class initialized with usn

while(1){

clrscr();

title();

cout<<"\tLogin type : Administrator ["<<id<<"]"<<endl;

cout<<"\t\tMenu"<<endl;

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

cout<<"\t\t 1 : Admit student\n\t\t 2 : Remove Student\n\t\t 3 : Add Staff\n\t\t 4 : Remove Staff\n\t\t 5 : List Students\n\t\t 6 : Manage attendance";

cout<<"\n\t\t 7 : View student attendance\n\t\t 8 : View student marks \n\t\t 9 : Notifications \n\t\t10 : Check Complaints \n\t\t11 : Exam Time Table \n\t\t12 : Exit"<<endl;

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

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

int choice;

cin>>choice;

char s[5];

switch(choice){

case 3: ad.addstaff();

break;

case 1: clrscr();

ad.admitStudent();

break;

case 2: cout<<"\n\tEnter the valid USN : ";

cin>>pass;

strupr(pass);

ad.removeStudent(pass);

break;

case 4: cout<<"\n\tEnter the staff id : ";

cin>>pass;

strupr(pass);

ad.removeStaff(pass);

break;

case 5: clrscr();

ad.listStudent();

break;

case 6:clrscr();

ad.manageAttendence();

break;

case 8: clrscr();

ad.viewMarks();

break;

case 7: clrscr();

ad.viewAttendence();

break;

case 9: clrscr();

ad.Notification();

break;

case 10: clrscr();

ad.complaint();

break;

case 11 : clrscr();

ad.timetable();

break;

case 12 : cout<<"Logging out...["<<id<<"]";

Sleep(1000);

return 0;

case 23 :

ad.showProfile();

default : cout<<"Select valid choice : ";

Sleep(1000);

}

}

}

int guest(){

fflush(stdin);

clrscr();

title();

char filename[40], ch;

ifstream read("topic.dat",ios::binary|ios::in);

if(!read.is\_open()){

return F\_NOT\_FOUND;

}

read.seekg(0);

cout<<"\n ";

fflush(stdin);

cout<<"TOPICS\n";

while(!read.eof()){

read.get(ch);

cout<<ch;

}

cout << "\nPlease enter paper name(as shown above, please use underscore instead of space): ";

cin>>filename;

strcat(filename,".dat");

cout<<filename<<endl;

ifstream read1;

read1.open(filename);

cin.get();

if(!read1.is\_open()){

cout<<"Please Enter a valid paper name!!";

cin.get();

return F\_NOT\_FOUND;

}

title();

read1.seekg(0);

cout<<"\n ";

while(!read1.eof()){

read1.get(ch);

cout<<ch;

}

cout<<"\nPress any key to return...";

cin.get();

read1.close();

read.close();

}

**SNAPSHOTS**

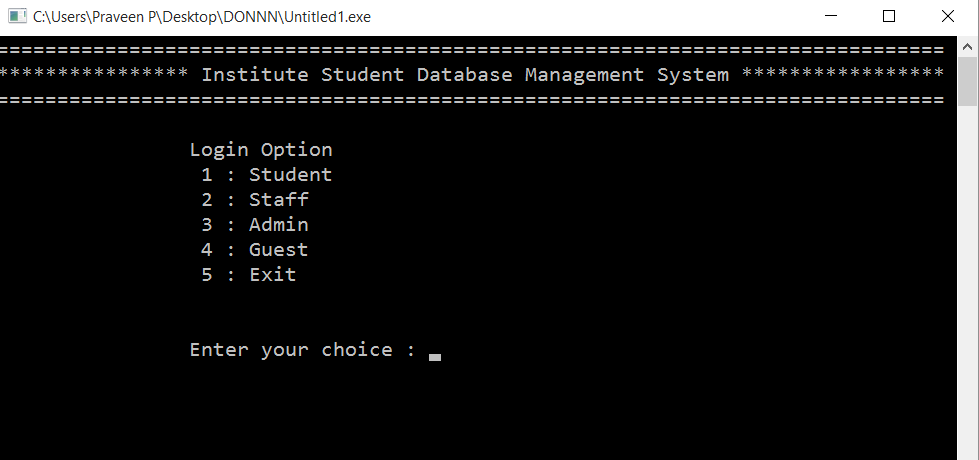
****

Fig1 : shows the login option.

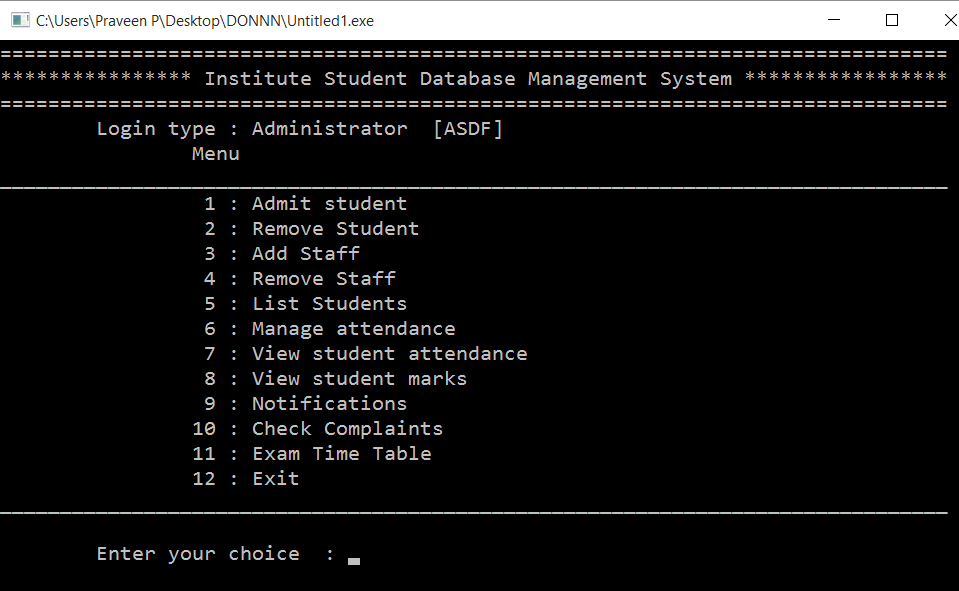
****

Fig2 : shows the admin interface.

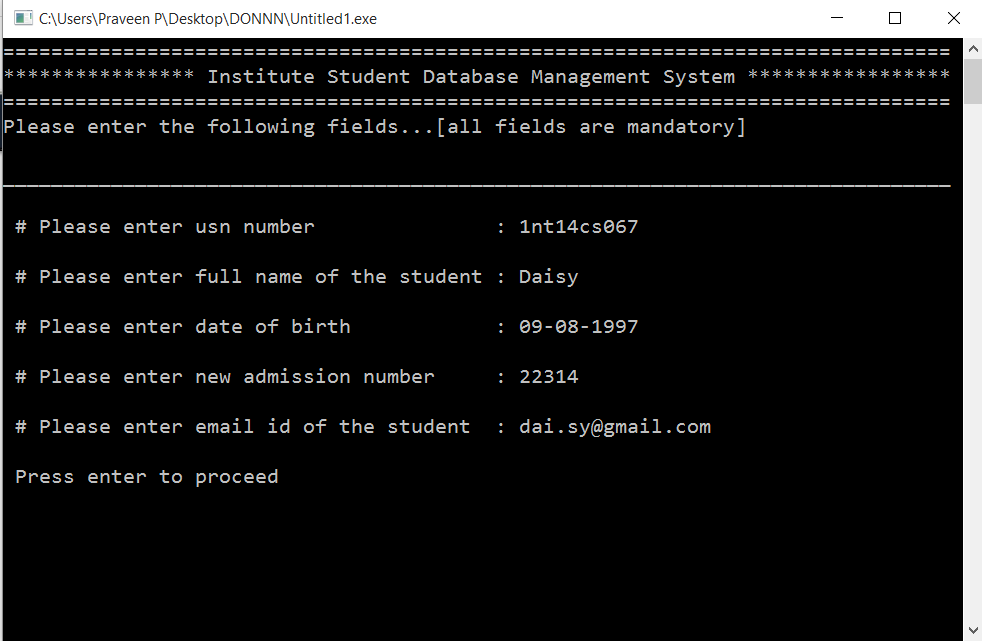
****

Fig3 : adding new student details.

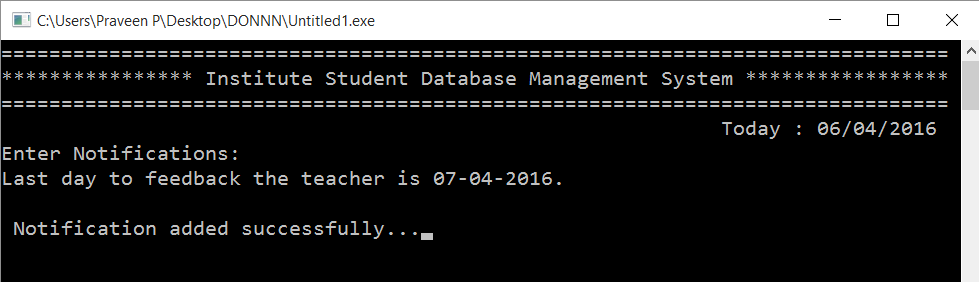
****

Fig4 : Enter notification

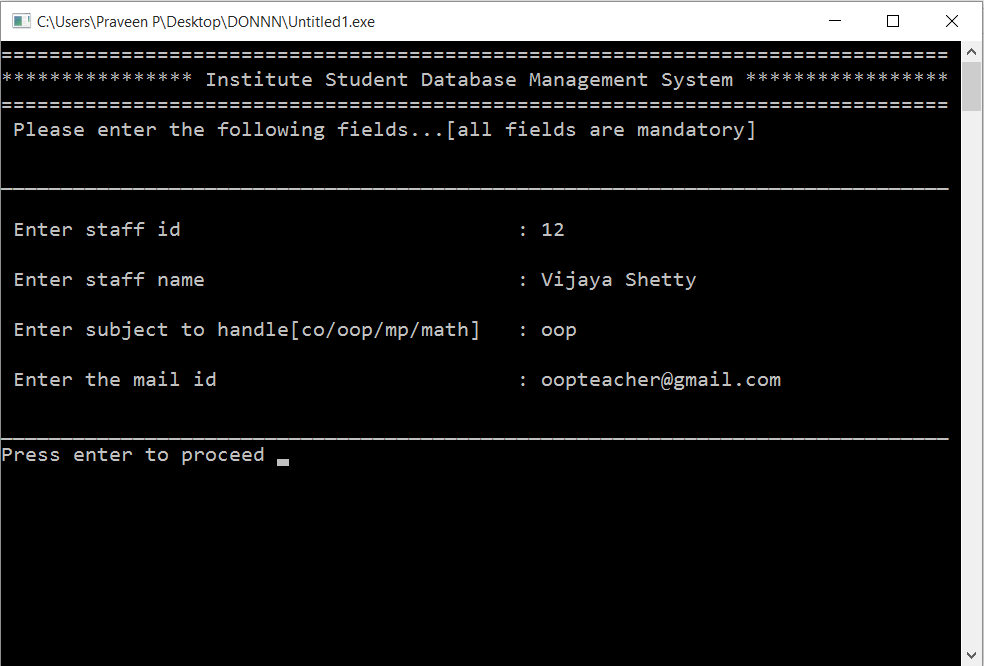


Fig5 : Add Staff

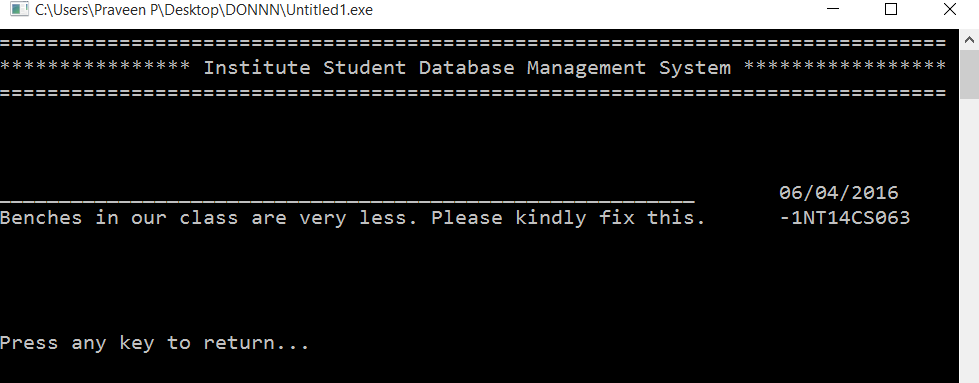


Fig6 : check Complaints

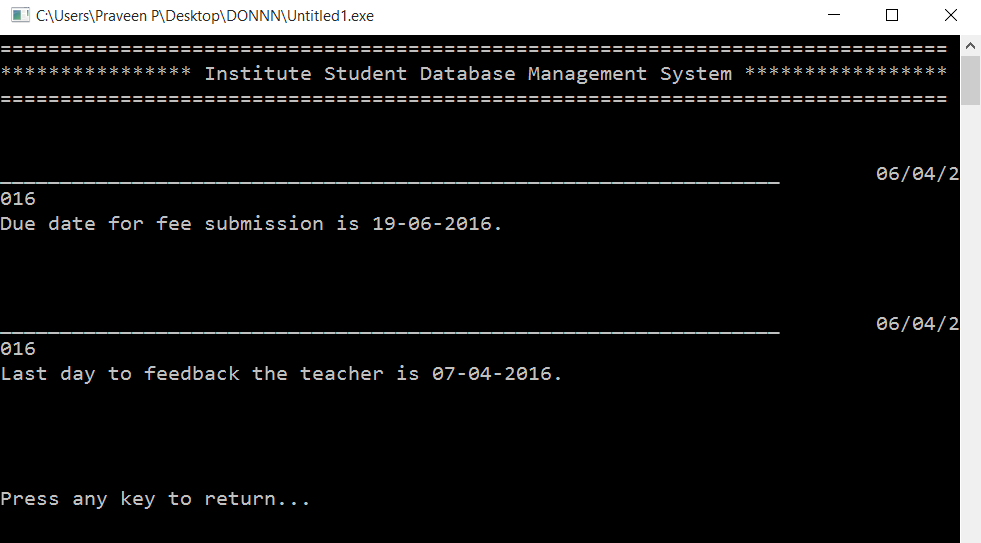


Fig7 : Notification view in student profile

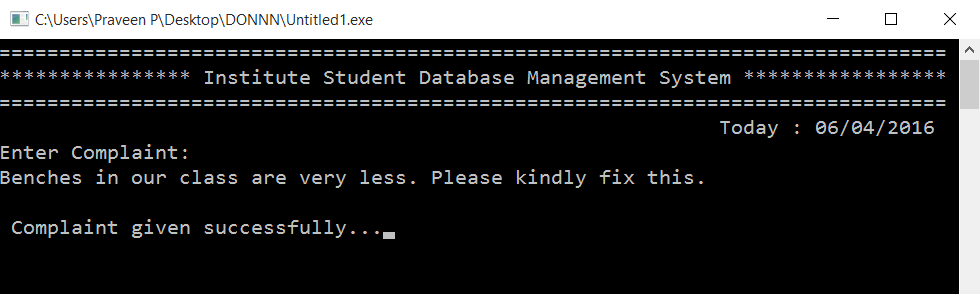


Fig 8: Enter Complaint

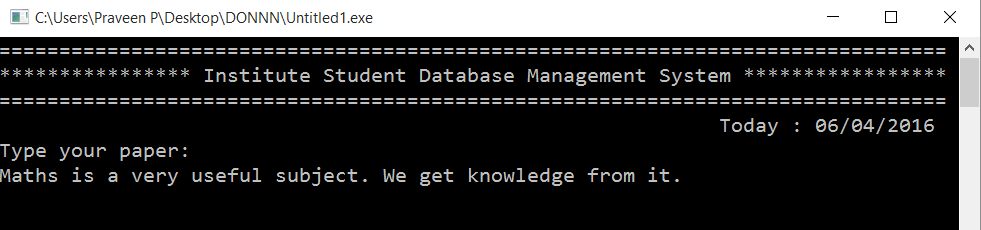


Fig 9: Enter Papers

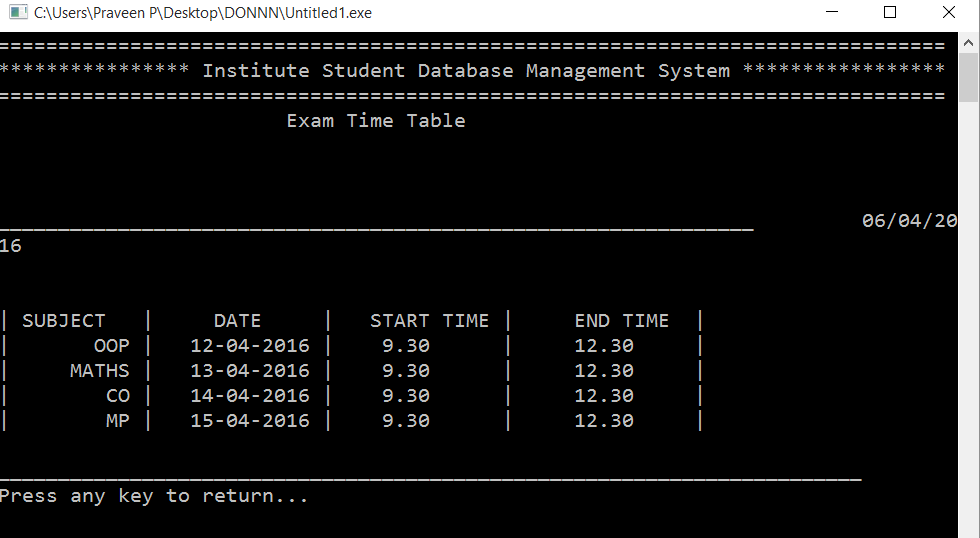


Fig 10: View TimeTable

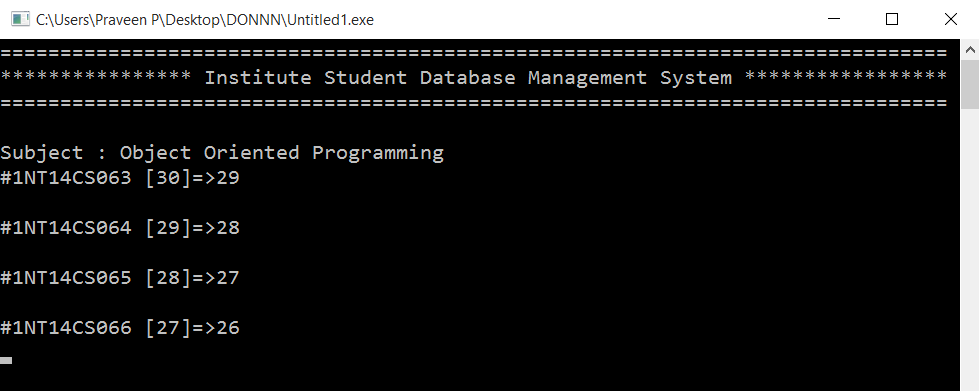


Fig 11: Enter Marks

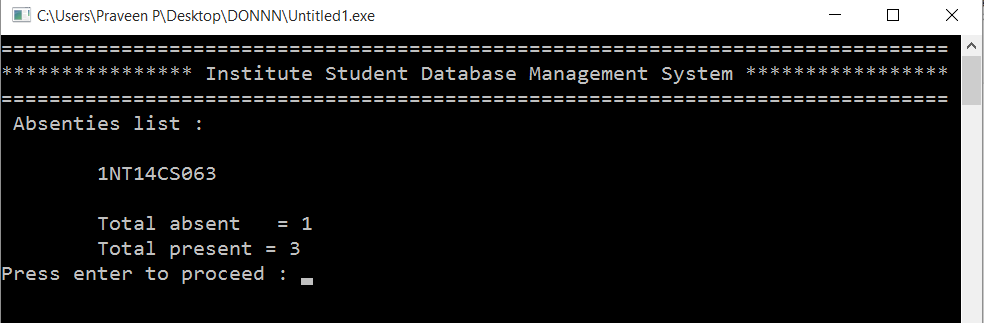


Fig 12: Enter attendance

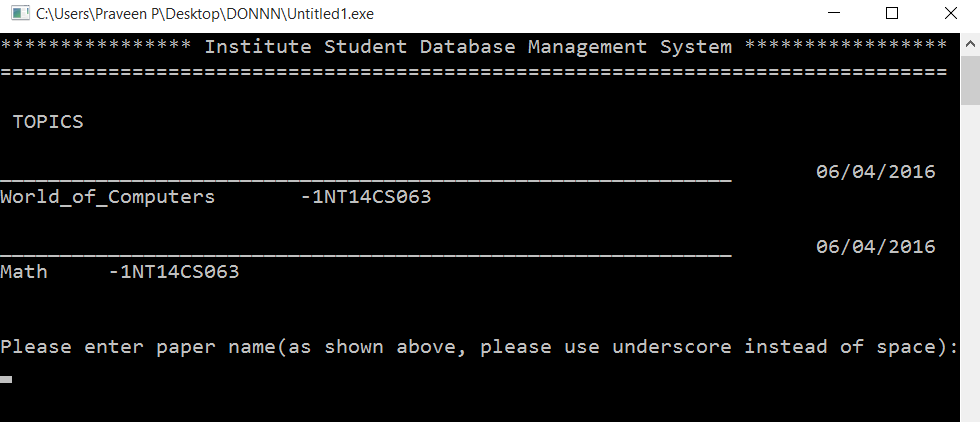


Fig13: View Papers List in Guest Mode

**CONCLUSION**

The project can be successfully implemented for student database management system. Manual work can be reduced to a great extent. A more systematic way of managing student database is implemented. Pen and paper work is abolished.

This project has helped us learn a lot of things. Working together in a team has helped us exchange our ideas manipulate and transform them into working programs. The successful completion of this project has given us a sense of satisfaction and built in us a level of confidence to handle jobs related to this course.

This project is successfully completed. And once again thank all the people who helped us in completing this project without which it would have not been successful.

Thanking you.

**BIBLIOGRAPHY**

* The C++ Programming Language (special edition) by Bjarne Stroustrup
* C++ The complete reference by Herbert Schildt
* [www.cs.cf.ac.uk](http://www.cs.cf.ac.uk)
* www.tutorialspoint.com