**P.R.I.S.M**

**Programmed Record keeping Interface for Student Management**

**ACKNOWLEDGEMENT**

First and foremost I would like to thank god almighty for his blessings that kept me energetic and involved throughout the making of this project.

I wish to extend my gratitude to Ms. Deepa Das, our principal and Mr. Vinod, our vice principal for giving us the opportunity to perform the requisite activities behind the successful completion of this project work.

I also wish to acknowledge my sincere respect and gratitude to my subject teacher Ms Neethu for giving us the valuable support and cooperation without which this endeavor would not have been possible.

Nevertheless I would like to thank my group members for all their contributions that materialized into this project and also for sharing the work load so we were able to complete the project within the limited time frame.

**ABSTRACT**

This project is based on the development of an advanced student management system called **P.R.I.S.M** – **P**rogrammed **R**ecord keeping **I**nterface for **S**tudent **M**anagement.

This student management software can be used by educational institutions of all scales for maintaining student records, keeping them up to date.

This management software is highly versatile and can be configured for use by both the school administration and public alike to obtain the information on a student.

Achieving the same task through conventional means is a waste of precious time and money on top of the fact that manual record keeping is redundant, almost impossible to update, modify and share information on the go.

**P.R.I.S.M** has been coded in **C++** using some of its basic features and primarily implements the concept of classes and objects. The versatile code relies on binary file handling techniques to store records.

**INDEX**

|  |  |
| --- | --- |
| **TOPIC** | **PAGE** |
| * Introduction | **1** |
| * System Design | **2** |
| + Module description | **3-4** |
| + Data Flow Diagram | **5-6** |
| + Data Design | **7** |
| * Hardware and Software used | **10** |
| * Source Code | **11-25** |
| * Result | **25-29** |
| * References | **31** |

**INTRODUCTION**

PRISM has friendly yet robust user interfaces which enhances its simplicity such that even the most untrained people can operate it and retrieve data with ease, guided by user friendly instructions along the way.

PRISM can keep records of students, storing and maintaining info on their marks, name, roll no., grade etc., in a binary file. The proposed logical system will keep the data centralized in a secure machine, while simultaneously allowing users and administrators to access data from their own individual workstations.

PRISM can be operated by both users and administrators so; the software provides two modes by which you can log in – as an administrator or as a guest (user).

Guests can go through and search records by roll number to get relevant details, while administrator logins will be prompted to enter their username and password combination and once logged in, will have the potential to add, edit, search, modify and delete records.

**SYSTEM DESIGN**

**A. Module Description**

**PRISM** makes the task of storing and retrieving student details efficient and easy. It currently supports but is not limited to the following features - adding new records, deleting records, modification of existing records, searching existing records, username-password based log ins’, etc.

* Module 1 : Splash Screen

This module will prompt the user to log in as an Administrator or a Guest user.

* Module 2 : Administrator Login

Administrators will be prompted to log in with correct username password combination

* Module 3 : Record Keeping
* **INSERTION: inserts a new record.**
* **DELETION : deletes an existing record.**
* **MODIFICATION : modifies an existing record.**
* **SEARCH : searches and displays records with roll no.**

ADDITIONAL MODULES

* **MODULE TO FIND GRADE**

**Finds grades of students based on marks entered.**

* **ERROR REPORTING MODULE**

**Displays error message and terminates the program if undesired input is entered by the user.**

* **USER INTERFACE MODULES**

**Displays title bar, loading animation etc.**

**Improves user friendliness.**

* **PASSWORD CHANGING MODULE**

**Facilitates password changing option for administrators**

**B. Data Flow Diagram**

**0 th Level**

STUDENT DETAILS QUERY

**.DAT FILE**

**USER**

RESULT REPORT

**1 st Level**

**GUEST MENU**

MODIFICATION

MODIFICATION

DISPLAY

**.DAT FILE**

**C. Data Design**

**Header Files Used In the Program**

* fstream.h - contains++stream classes that support file input/output
* conio.h - contains several useful library functions for console I/O.
* stdio.h - defines types and macros needed for standard I/O package.
* string.h - contains several string manipulation functions.
* process.h - contains functions used to work with threads & processes
* ctype.h - contains character identification/manipulation functions
* stdlib.h - standard c++ library functions
* dos.h - contains many useful functions such as delay()

**File and Variables Used In the Program**

**studdata.dat:**The file stores the records on students and contains information such as -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SNO | FIELD NAME | TYPE | SIZE | FIELD DESCRIPTION |
| 1 | Name | char | 25 bytes | Student name |
| 2 | Perc | float | 4 bytes | Percentage secured |
| 3 | Roll | Int | 25 bytes | Roll no. of student |
| 4 | Grade | char | 1 byte | Grade of each student |
| 5 | Mark | int | 2 bytes | Marks secured |

**HARDWARE AND SOFTWARE REQUIREMENTS**

**Hardware specifications**

* Pentium IV-2GHz
* Microsoft Windows XP
* 512MB RAM
* 30 MB free hard disk space (minimum)

**Software Used**

* Frontend: TURBO C++
* Backend : Data Files

**SOURCE CODE**

#include<dos.h>

#include<fstream.h>

#include<conio.h>

#include<string.h>

#include<ctype.h>

#include<stdlib.h>

#include<stdio.h>

#include<process.h>

int secure=1;

void MAIN();

void sdel();

void intro();

void logout();

void titleonly();

void titleandloading();

void adminlogin();

class student

{

int roll;

char NAME[20],grade;

float mark[5],perc;

public:

void get();

void disp();

void grader();

int retroll()

{

return roll;

}

};

void student::get()

{

cout<<"\nEnter the name of the student : ";

gets(NAME);

cout<<"\nEnter the roll number : ";

cin>>roll;

cout<<"\nEnter the mark of 5 subjects\n";

for(int i=0;i<5;i++)

{cin>>mark[i];

cout<<"\t";}

}

void student::disp()

{

cout<<"\n\n";

cout<<"\nRoll number : "<<roll;

cout<<"\nName : ";puts(NAME);

cout<<"Marks are : \n";

for(int i=0;i<5;i++)

cout<<"MARK"<<i+1<<" "<<mark[i]<<"\n";

grader();

cout<<"\nPercentage : "<<perc;

cout<<"\t\tgrade : "<<grade;

}

void student::grader()

{

perc=0;

for(int i=0;i<5;i++)

perc+=mark[i];

perc=perc/5;

if(perc>=90)grade='A';

if(perc>=80)grade='B';

if(perc>=70)grade='C';

if(perc>=50)grade='D';

if(perc>=40)grade='E';

else

grade='F';

}

void writerec()

{

student ob;

ofstream f("A.dat",ios::app|ios::binary);

char ch='y';

while(ch=='y')

{

ob.get();

clrscr();

cout<<"\t\tchanges written";

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

cout<<"\nDo you want to add more ? (y/n)\n";

cin>>ch;

}

f.close();

if(ch=='n')

{

MAIN();}

getch();

}

void searchrec()

{

student ob;

char ch;

int k,i=0;

fstream f;

f.open("A.dat",ios::binary|ios::in|ios::out);

cout<<"\n\t\t\t\tsecure->";cout<<::secure;

cout<<"\n\n Enter the roll no of the student ";

cin>>k;

while(f.read((char\*)&ob,sizeof(ob))&&i==0)

{

if(ob.retroll()==k)

{ i++;

cout<<"\n\n\n Match found";

ob.disp();getch();

if(secure==1){MAIN();}else intro();

}

}

f.close();

if(i==0)

{cout<<"\n No match found.Please check the roll number entered";

getch();

if(secure==1){MAIN();}else intro();}

}

void modify()

{

student ob;

char ch;

int k,found=0;

fstream f;

f.open("A.dat",ios::binary|ios::in|ios::out);

cout<<"\n\t\t\t\tsecure->";cout<<::secure;

cout<<"\n\n Enter the roll no of the student ";

cin>>k;

while(f.read((char\*)&ob,sizeof(ob)))

{

if(ob.retroll()==k)

{ cout<<"\n\n\n Match found\n";

cout<<"enter new details\n";

ob.get();

int pos=f.tellg()-sizeof(ob);

f.seekp(pos);

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

found=1;

break;

}

}

f.close();

if(found==0)

{

cout<<"\n No match found.Please check the roll number entered";

getch();

MAIN();

}

else

{

cout<<"Record Modified....";

getch();

MAIN();

}

}

void sdel()

{

student ob;

char ch;

int k,found=0;

ofstream fout;

ifstream fin;

fout.open("t.dat",ios::binary);

fin.open("A.dat",ios::binary);

cout<<"\n\t\t\t\tsecure->";cout<<::secure;

cout<<"\n\n Enter the roll no of the student ";

cin>>k;

while(fin.read((char\*)&ob,sizeof(ob)))

{

if(ob.retroll()!=k)

fout.write((char\*)&ob,sizeof(ob));

else

found=1;

}

fin.close();

fout.close();

if(found==0)

{

cout<<"\n No match found.Please check the roll number entered";

getch();

MAIN();

}

else

{

cout<<"Record deleted....";

getch();

remove("A.dat");

rename("t.dat","A.dat");

MAIN();

}

}

void intro();

void logout();

void titleonly()

{

clrscr();

textattr(BLACK);

textbackground(WHITE);

cprintf(" PRISM BETA V.1.0 "); textattr(RED+BLINK);textbackground(WHITE);

cprintf(" ONLINE "); textattr(YELLOW);

cprintf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

textattr(WHITE);

}

void error()

{clrscr();cout<<"\n\n\n\t\t\t\tFATAL ERROR";getch();exit(0); }

char uname[20],pass[20];

char \*name="defaultname";

char \*passu="defaultpass";

void titleandloading();

void MAIN()

{

titleonly();

cout<<"\n\t\tWELCOME "<<name<<"\n\n\t\t";

cout<<"\n\n\t\t\t1.Change admin name and pass\n\n\t\t\t2.Search database\n\n\t\t\t3.Modify database\n\n\t\t\t4.Add new entry\n\n\t\t\t5.Mainmenu\n\n\t\t\t6.Delete\n\n";

int choice;

cin>>choice;

if(choice==6)

{logout();}

if(choice==1)

{clrscr();

titleonly();

cout<<"\n\n\n\n\n\t\t";

cout<<"Enter new admin name : ";

cin>>::name;

cout<<"\n\n\n\t\t";

cout<<"Enter new admin pass : ";

cin>>::passu;

cout<<"\n\n\tCHANGES SUCCESSFUL";

getch();

logout();

}

if(choice==5)

{

intro();

}

if(choice==3)

{ clrscr();cout<<"\n\n\n";titleonly();

modify();

}

if(choice==2)

{ clrscr();cout<<"\n\n\n";titleonly();

searchrec();

}

if(choice==4)

{ clrscr();cout<<"\n\n\n";titleonly();

writerec();

}

if(choice==7)

{ clrscr(); cout<<"\n\n\n";titleonly();

sdel();

}

else

error();

}

void loading()

{

textattr(YELLOW);

{delay(245);

cprintf("");

delay(240);

cprintf("");

delay(245);

cprintf("");

delay(240);

cprintf("");

delay(245);

cprintf("");

delay(240);

cprintf("");

delay(245);

cprintf("");

delay(240);

cprintf("");

delay(245);

cprintf("");

}

delay(1000);

textattr(WHITE);

}

void intro();

void credits()

{titleandloading();

cout<<"\n-------------------------------------------------------------------------------\n";

cout<<"\t\t\t\t CREDITS\n";

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

textattr(LIGHTMAGENTA);

delay(1000);

textattr(YELLOW+BLINK);

cout<<"\n";cprintf(" P");

delay(450);cout<<"rogrammed";

delay(450);textattr(RED+BLINK);

cprintf(" R");delay(450);

cout<<"ecord keeping";

delay(450);textattr(GREEN+BLINK);

cprintf(" I");delay(450);cout<<"nterface for";

delay(450);textattr(LIGHTRED+BLINK);cprintf(" S");

delay(450);cout<<"tudent ";delay(450);

textattr(LIGHTCYAN+BLINK);

cprintf("M");delay(450);

cout<<"anagement\n";textattr( LIGHTMAGENTA);

delay(700);

cout<<"\n\t\t\t ";

cout<<"\n\n";

textattr(WHITE);

cout<<"\n\n\n\n CLASS 12 2017-18 ";

textattr(LIGHTGREEN);

cprintf(";)");

textattr(WHITE);

cout<<"\t\t #EPIC\_PROJECT\t\t\t #BEAT\_THAT !\n";

textattr(RED);

cprintf("…………………………………………………………………………………………………...…”);

cout<<"\t\t\t\t ";

textattr(WHITE);

getch();

intro();

}

void titleandloading()

{

clrscr();

textattr(BLACK);

textbackground(WHITE);

cprintf(" PRISM BETA V.1.0 "); textattr(RED+BLINK);textbackground(WHITE);

cprintf(" ONLINE "); textattr(WHITE);

loading();

}

void exitquote()

{

textattr(LIGHTGREEN);

clrscr();

cout<<"\n\n\n\n\n\n\n\t\t\t";

cprintf(" STAY HUNGRY , STAY FOOLISH");

textattr(WHITE);

cout<<"\n\n\t\t\t\t";cprintf(" -Steve Jobs");

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

textattr(CYAN);

cout<<"\n\n\n\n\n\t\t ";cprintf("Made with TURBOC++ and a lot of time...! ");

delay(3000);exit(0);

}

void intro()

{

int x;

textattr(WHITE);

titleandloading();

cout<<"--------------------------------------------------------------------------------\n\n";

cout<<"\n\n\n\t\t\t\t1. ADMIN LOGIN\n\n\n\n\t\t\t\t2. GUEST LOGIN\n\n\n\n\t\t\t\t3. CREDITS\n\n\n\n\t\t\t\t4. EXIT\n";

cin>>x;

switch (x)

{

case 1:

adminlogin();

break;

case 2:

::secure=0;

clrscr();

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

titleonly();

searchrec();

break;

case 3:

credits();

break;

case 4:

exitquote();

break;

default: error();

}

}

void adminlogin()

{

titleonly();

cout<<"\n\t\t\t\tADMINISTRATOR LOGIN";

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

cout<<"\tUSERNAME: ";

gets(uname);

cout<<"\n\tPASSWORD: ";

gets(pass);

cout<<"\n";

loading();

cout<<"\n";

if((strcmp(uname,name)==0)&&(strcmp(pass,passu)==0))

{ ::secure=1;

cout<<"\n\t\tLOGIN SUCCESSFUL\n\t\t\t\t\tsecure->1";

cout<<"\n\n\t\tPRESS ENTER\n";

getch();

clrscr();

MAIN();}

else

{

clrscr();

cout<<"\n\n\n\t\t OOPS! PASSWORD AND USERNAME DIDNT' MATCH, \n\n\n\t\t ENTER ANY KEY TO EXIT, ENTER 'T'TO TRY AGAIN \n\t\tOR 'R' TO RETURN TO MAINMENU\n";

char t;

cin>>t;

if(t=='T')

adminlogin();

else if(t=='R')

intro();

else

exit(0);

}

}

void logout()

{

intro();

}

void main()

{

clrscr();

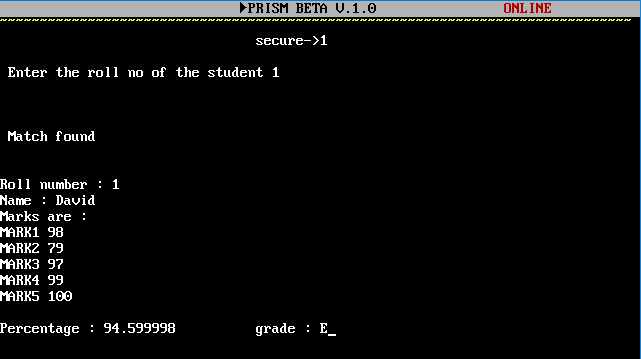
intro();

getch();

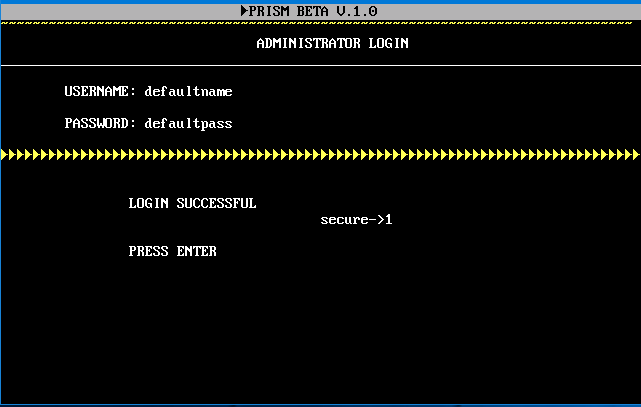
}

**RESULT**

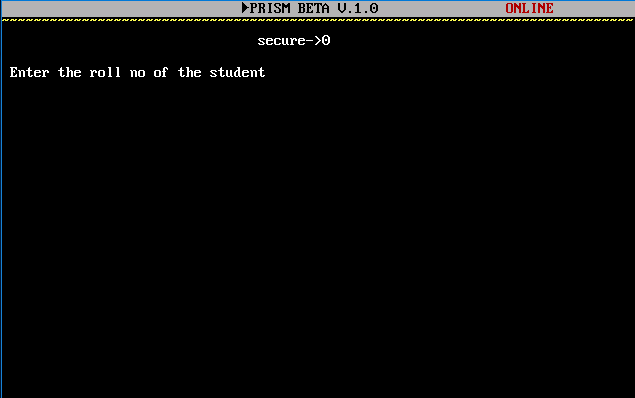
**search function**



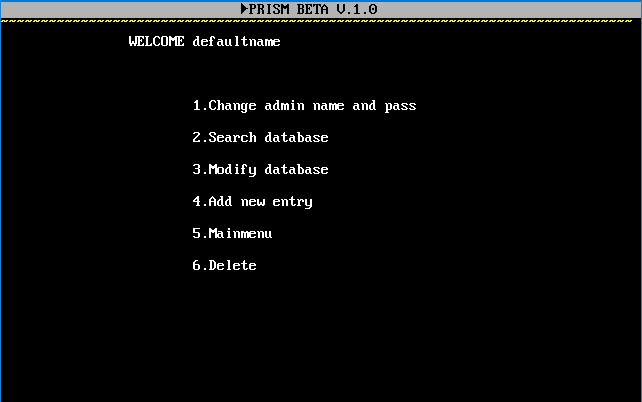
**admin login screen**



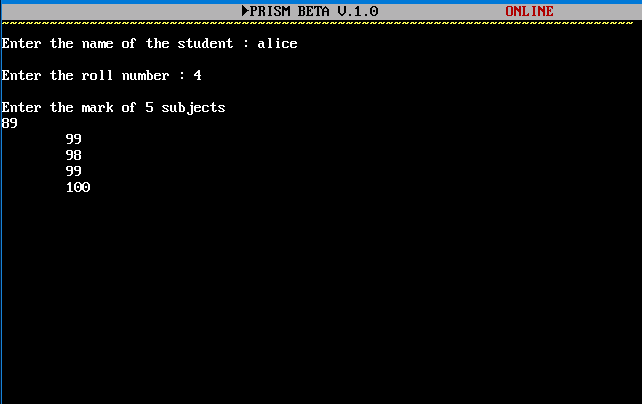
**guest panel**



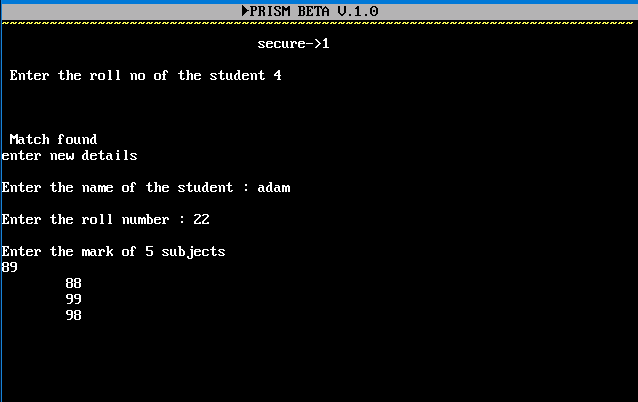
**admin panel**

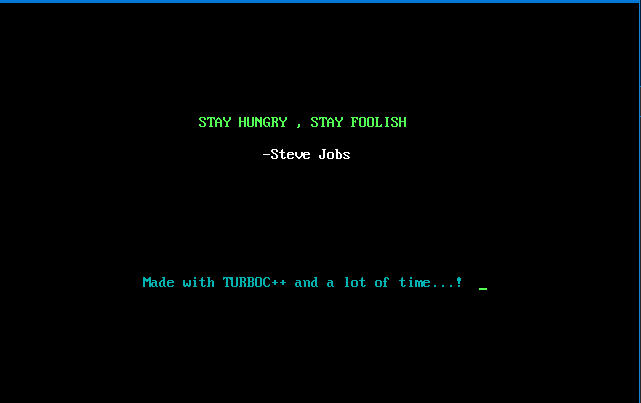


**insersion**



**Modification**



**exit splash screen** 

**CONCLUSION AND FUTURE ENHANCEMENTS**

We believe the project has successfully been completed. It is capable of fulfilling the basic necessities of a student management software on demand. Core features of c++ such as classes, objects, binary files etc have been successfully implemented in the program which makes it less complex , more user-friendly and understandable.

This project shed some light on general OOP concepts and modeling of real world system. The usage of different inbuilt modules in the c++ library demonstrate the ability to develop and derive new classes, structures ,functions and organize them such that they will model real world systems within computers.

Some changes can be made as to enhance the performance of the program under various circumstances. the program presently uses a simple design concept. The system can be made more comprehensive by the integration of a school library management by linking the current database with that of a library or something similar. The program had to be developed within a narrow margin of time hence it could not be advanced further as the topic allowed.

**REFERENCE**

* Computer Science with C++ XII, Sumita Arora