INSTRUCTIONS:

1. The project report should be neatly typed.
2. Avoid using Abbreviations.
3. The text should be justified and typed in the Font style ‘Times New Roman’ and Font size ‘12’.
4. Heading and subheading should be bold.
5. The length of the report may be about 10 to 15 pages.

Sample Cover Page

DON BOSCO INSTITUTE OF TECHNOLOGY

Skill Lab: C++ and Java Programming
MINI PROJECT REPORT

On

"Title of mini-project"
2021-22

Submitted By:

Yash Sonavane 45

Under the guidance of
Ms. Deepali Kayande

Inner Title Page

Mini Project Title : Student Management System

Institute Name : Don Bosco Institute of Technology.

Institute Address : Premier Automobiles Road,
Kurla (West), Mumbai – 400070

Department : EXTC

Class : _____

Project Group Members :

	Names of students	Roll No.
1.	Yash Sonavane	45
2.	Sanskar Kumar	
3.	Suraj Das	42
4.	Tejas Rokade	43

Date of Submission :15 December 2021

Guide : Ms. Deepali Kayande

TABLE OF CONTENTS

SR. NO.	CONTENT	PAGE NO.
CHAPTER 1	INTRODUCTION	
CHAPTER 2	PROBLEM DEFINITION MODULES (IF ANY)	
CHAPTER 3	IMPLEMENTATION	
CHAPTER 4	RESULTS(SNAPSHOTS)	
CHAPTER 5	CONCLUSION	
CHAPTER 6	REFERENCES	

CHAPTER 1

INTRODUCTION

This system saves the time of the student and of the administrator. It includes processes like registration of the student's details, assigning the department based on their course, and maintenance of the record. This system reduces the cost and workforce required for this job. As the system is online the information is globally present to everyone.

This makes the system easy to handle and feasible for finding the omission with updating at the same time. As for the existing system, they use to maintain their record manually which makes it vulnerable to security. If filed a query to search or update in a manual system, it will take a lot of time to process the query and make a report which is a tedious job.

CODE:

```
#include <iostream>
#include <fstream>
#include <iomanip>
#include <stdlib.h>

using namespace std;

class Student
{
int admno;
char name[20];
char gender;
int std;
float marks;
float percentage;

public:
void getData();
void showData();
int getAdmno() { return admno; }
} s;

void Student::getData()
{
cout << "\n\nEnter Student Details.....\n";
cout << "Enter Admission No.     : ";
cin >> admno;
cout << "Enter Full Name         : ";
cin.ignore();
cin.getline(name, 20);
cout << "Enter Gender (M/F)      : ";
cin >> gender;
```

```
cout << "Enter Standard         : ";
cin >> std;
cout << "Enter Marks (out of 500): ";
cin >> marks;
cout << endl;
percentage = marks * 100.0 / 500.00;
}

void Student::showData()
{
cout << "\n\n.......Student Details......\n";
cout << "Admission No.     : " << admno << endl;
cout << "Full Name         : " << name << endl;
cout << "Gender            : " << gender << endl;
cout << "Standard          : " << std << endl;
cout << "Marks (out of 500): " << marks << endl;
cout << "Percentage        : " << percentage << endl;
cout << endl;
}

void addData()
{
ofstream fout;
fout.open("Students.dat", ios::binary | ios::out | ios::app);
s.getData();
fout.write((char *)&s, sizeof(s));
fout.close();
cout << "\n\nData Successfully Saved to File....\n";
}

void displayData()
{
ifstream fin;
fin.open("Students.dat", ios::in | ios::binary);
while (fin.read((char *)&s, sizeof(s)))
```

```
{
s.showData();
}
fin.close();
cout << "\n\nData Reading from File Successfully Done....\n";
}

void searchData()
{
int n, flag = 0;
ifstream fin;
fin.open("Students.dat", ios::in | ios::binary);
cout << "Enter Admission Number you want to search : ";
cin >> n;

while (fin.read((char *)&s, sizeof(s)))
{
if (n == s.getAdmno())
{
cout << "The Details of Admission No. " << n << " shown herewith:\n";
s.showData();
flag++;
}
}
fin.close();
if (flag == 0)
cout << "The Admission No. " << n << " not found....\n\n";
cout << "\n\nData Reading from File Successfully Done....\n";
}

void deleteData()
{
int n, flag = 0;
ifstream fin;
ofstream fout, tout;
```

```
fin.open("Students.dat", ios::in | ios::binary);
fout.open("TempStud.dat", ios::out | ios::app | ios::binary);
tout.open("TrashStud.dat", ios::out | ios::app | ios::binary);

cout << "Enter Admission Number you want to move to Trash : ";
cin >> n;

while (fin.read((char *)&s, sizeof(s)))
{
if (n == s.getAdmno())
{
cout << "The Following Admission No. " << n << " has been moved to
Trash:\n";
s.showData();
tout.write((char *)&s, sizeof(s));
flag++;
}
else
{
fout.write((char *)&s, sizeof(s));
}
}
fout.close();
tout.close();
fin.close();
if (flag == 0)
cout << "The Admission No. " << n << " not found....\n\n";
remove("Students.dat");
rename("tempStud.dat", "Students.dat");
}

void getTrash()
{
ifstream fin;
```

```
fin.open("TrashStud.dat", ios::in | ios::binary);
while (fin.read((char *)&s, sizeof(s)))
{
s.showData();
}
fin.close();
cout << "\n\nData Reading from Trash File Successfully Done....\n";
}

void modifyData()
{
int n, flag = 0, pos;
fstream fio;

fio.open("Students.dat", ios::in | ios::out | ios::binary);

cout << "Enter Admission Number you want to Modify : ";
cin >> n;

while (fio.read((char *)&s, sizeof(s)))
{
pos = fio.tellg();
if (n == s.getAdmno())
{
cout << "The Following Admission No. " << n << " will be modified with
new data:\n";
s.showData();
cout << "\n\nNow Enter the New Details....\n";
s.getData();
fio.seekg(pos - sizeof(s));
fio.write((char *)&s, sizeof(s));
flag++;
}
}
fio.close();
```

```
if (flag == 0)
cout << "The Admission No. " << n << " not found....\n\n";
}

void project()
{
 int ch;
 do
 {
system("cls");
cout << ".............STUDENT MANAGEMENT SYSTEM.............\n";
cout <<
"=================================================\n";
cout << "0. Exit from Program\n";
cout << "1. Write Data to File\n";
cout << "2. Read Data From File\n";
cout << "3. Search Data From File\n";
cout << "4. Delete Data From File\n";
cout << "5. Get Deleted Records from Trash file\n";
cout << "6. Modify Data in File\n";
cout << "Enter your choice  : ";
cin >> ch;
system("cls");
switch (ch)
 {
case 1:
  addData();
 break;
case 2:
displayData();
 break;
case 3:
searchData();
 break;
```

```
                case 4:
                 deleteData();
                     break;
                case 5:
                  getTrash();
                     break;
                case 6:
                modifyData();
                     break;
                    }
          system("pause");
        } while (ch);
          }

    int main()
          {
system("color B0");
          project();
          }
```

RESULTS:

C:\Users\Yash\Downloads\Lab 7 Reference Programs-20211116\Untitled1.exe

```
..............STUDENT MANAGEMENT SYSTEM.............
=====================================================
0. Exit from Program
1. Write Data to File
2. Read Data From File
3. Search Data From File
4. Delete Data From File
5. Get Deleted Records from Trash file
6. Modify Data in File
Enter your choice  :
```

C:\Users\Yash\Downloads\Lab 7 Reference Programs-20211116\Untitled1.exe

```
Enter Student Details......
Enter Admission No.     : Enter Full Name        : Enter Gender (M/F)      : Enter Standard         : Enter Marks (out of 500):


Data Successfully Saved to File....
Press any key to continue . . .
```

CONCLUSION:

Student Management System can be used by educational institutions to maintain their student records easily. Achieving this objective is difficult using the manual system as the information is scattered, can be redundant, and collecting relevant information may be very time-consuming. All these problems are solved by this project.

This system helps in maintaining the information of pupils of the organization. It can be easily accessed by the manager and kept safe for a long period of time without any changes

REFERENCE:

https://www.lovelycoding.org