Bangladesh University of Business and Technology

(BUBT)



Project Report

Project Name: Students Management System

Course Title : Software Development 1

Course Code : CSE 100

Submitted to:

Md. Mahbub-Or-Rashid

Assistant Professor

Department of CSE

Bangladesh University of Business and Technology (BUBT)

Team Member :

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name*** | ***ID*** | ***Intake*** | ***Section*** |
| ***Md Takiul Islam Sumon*** | ***22234103276*** | ***50*** | ***05*** |
| ***Moinul Islam Rehan*** | ***22234103367*** | ***50*** | ***05*** |
| ***Md. Mursaline Ahmed*** | ***22234103278*** | ***50*** | ***05*** |
| ***Mainul Wazid*** | ***22234103271*** | ***50*** | ***05*** |
| ***Md. Mortaza Ali*** | ***22234103274*** | ***50*** | ***05*** |

***Table of Contents***

[Abstract 4](#_Toc50273)

[Acknowledgement 5](#_Toc50274)

[Declaration 6](#_Toc50275)

[Copyright 7](#_Toc50276)

[Certificate 8](#_Toc50277)

[Approval 9](#_Toc50278)

[Dedication 10](#_Toc50278)

[Chapter 1 11](#_Toc50279)

[1.1 Introduction 11](#_Toc50280)

[1.2 Purpose: 11](#_Toc50281)

[1.3 Scope: 12](#_Toc50282)

[1.4 Project Overview: 12](#_Toc50283)

[Chapter 2 13](#_Toc50284)

[2.1 System Analysis: 13](#_Toc50285)

[2.2 Existing System Analysis: 13](#_Toc50286)

[2.3 Proposed system: 14](#_Toc50287)

[2.4 Feasibility Study: 14](#_Toc50288)

[2.4.1 Operation Feasibility: 15](#_Toc50289)

[2.4.2 Technical Feasibility: 15](#_Toc50290)

[2.4.3 Economical Feasibility: 15](#_Toc50291)

[Chapter: 3 16](#_Toc50292)

[3.1 Requirement analysis of student management system 16](#_Toc50293)

[3.2 Software Requirements of student management system 16](#_Toc50294)

[Chapter 4 17](#_Toc50295)

[4.1 Software: 17](#_Toc50296)

[4.1.1 Code::Blocks 17](#_Toc50297)

[4.2 Programming Language 18](#_Toc50298)

[4.2.1 C Language 18](#_Toc50299)

[Chapter 5 19](#_Toc50300)

[5.1 Architectural Design 19](#_Toc50303)

[5.1.1 Flowchart 19](#_Toc50304)

[5.1.2 Sub-Features 20](#_Toc50305)

[Chapter 6 22](#_Toc50311)

[6.1 System over View: 22](#_Toc50312)

[6.1.1. Loading: 22](#_Toc50313)

[6.1.2. Welcome Message: 22](#_Toc50314)

[6.2 Admin Module: 23](#_Toc50315)

[6.3 Menu: 23](#_Toc50316)

[6.3.1 Add Student: 24](#_Toc50317)

[6.3.2 Modify Student: 24](#_Toc50318)

[6.3.3 Show all student: 25](#_Toc50319)

[6.3.4 Individual View: 25](#_Toc50320)

[6.3.5 Delete: 26](#_Toc50321)

[Chapter 7 27](#_Toc50330)

[7.1 Code Analysis: 27](#_Toc50331)

[Chapter 8 43](#_Toc50332)

[8.1 REFRENCES 43](#_Toc50333)

[Chapter 9 44](#_Toc50332)

[9.1 Future Plan 44](#_Toc50333)

[Chapter 10 45](#_Toc50332)

[10.1 Conclusion 45](#_Toc50333)

***Abstract***

Student management system is a model system to store information about student of the city public schools like teacher profile, student records, classes, subjects, fees, timetables, datasheets and reports. The system is designed to meet the purpose of dealing with student information system.

As project mainly concentrate on Student management system so keeping the friendly user interface the system should provide all necessary Student information facilities. A Login Form which asks the User to browse the whole system and perform different operations step by step such as Saving, Updating, Deleting and loading records as well as providing the facility to Administrator to generate the sequence by hiding un-necessary tasks from the user. Also the system is capable of managing records for all the students. The system is also capable to generate reports about student results, classes and certificates.

Our project explains about the student management. This project mainly explains the various actions related to student details. This project shows some ease in adding, editing and deleting the student details. It also provides a less time consuming process for viewing, adding, editing and deleting the marks of the students.

# *Acknowledgement*

We like to say our gratitude to our creator ALLAH to let us into the world and our parents, who supported us in this whole study and always prayed for our success and good health. We express our deep sense of gratitude to our **Md. Mahbub-Or-Rashid** for his expert guidance stimulating discussions as well as continued impetus throughout the period of this project And endless patience towards the completion of this project. We feel very proud to work with him. Without the inspiring enthusiasm and encouragement of our supervisor, this work could not have been completed. We thank all the staffs and graduate students at **Bangladesh University of Business and Technology (BUBT)** and all the friends for their support and encouragement. We would also like to extend our elder and younger brothers. Finally, we wish to express our gratitude to **Bangladesh University of Business and Technology (BUBT)** for providing an excellent environment for research and all the other facilities to complete the project successfully.

***With best regards,***

**Md. Takiul Islam Sumon - 22234103276**

**Moinul Islam Rehan - 22234103367**

**Md. Mursaline Ahmed - 22234103278**

**Mainul Wazid - 22234103271**

**Md. Mortaza Ali - 22234103274**

# *Declaration*

We hereby declare that the project entitled Student Management System submitted in partial fulfillment of the requirements for the degree of *Bachelor of Science in Computer Science and Engineering* of **Bangladesh University of Business and Technology (BUBT)** is our own work and that it contains no material which has been accepted for the award to the candidate(s) of any other degree or diploma, except where due reference is made in the text of the project. To the best of our knowledge, it contains no materials previously published or written by any other person except where due reference is made in the project.

Md. Takiul Islam Sumon Moinul Islam Rehan Md. Mursaline Ahmed Mainul Wazid Md. Mortaza Ali

22234103276 22234103367 22234103278 22234103271 22234103274

# *Copyright*

*© Copyright by*

**Md. Takiul Islam Sumon, Moinul Islam Rehan, Md. Mursaline Ahmed, Mainul Wazid & Md. Mortaza Ali**

*2023 All Rights Reserved*

# 

# *Certificate*

This is to certify that the project work titled “**Students Management System”** and submitted by **Md. Takiul Islam Sumon(22234103276), Moinul Islam Rehan(22234103367), Md. Mursaline Ahmed(22234103278), Mainul Wazid(22234103271) & Md. Mortaza Ali(22234103274)**, in partial fulfillment for the requirement of embodies the work done by them under my supervision.

———————————————————

***Supervisor***

**Md. Mahbub-Or-Rashid**

Assistant Professor

*Department of Computer Science and Engineering*

**Bangladesh University of Business and Technology (BUBT)**

# *Approval*

The project work entitled **Student Management System** is submitted by **Md. Takiul Islam Sumon(22234103276), Moinul Islam Rehan(22234103367), Md. Mursaline Ahmed(22234103278), Mainul Wazid(22234103271) & Md. Mortaza Ali(22234103274)**under the Department of Computer Science and Engineering of **Bangladesh University of Business and Technology** **(BUBT)** is accepted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering.

***Supervisor***

**Md. Mahbub-Or-Rashid**

Assistant Professor

*Department of Computer Science and Engineering* **Bangladesh University of Business and Technology (BUBT)**

***Chairman***

**Md. Saifur Rahman**

*Department of Computer Science and Engineering* **Bangladesh University of Business and Technology (BUBT)**

# *Dedication*

*Our Loving Parents and Teachers who’s Support give us Strength and determination to accomplish our Goal…!!*

## Chapter 1

### 1.1 Introduction

Student Management System deals with all kind of student details, academic related reports, university details, batch details and other resource related details too. It tracks all the details of a student from the day one to the end of his course which can be used for all reporting purpose, completed semesters result, CGPA etc. Our design can facilitate us to explore all the activities happening in the university, the current status of a student. The student management system is an automated version of manual Student Management System. It can handle all details about a student. The details include students name, department, completed course, complete semester, id, CGPA. In case of manual system they need a lot of time, manpower etc. Here almost all work is computerized. So the accuracy is maintained. Maintaining backup is very easy. It can do with in a few minutes. Our system has three type of accessing modes, administrator, and student. Student management system is managed by an administrator. It is the job of the administrator to insert update and monitor the whole process. When an admin login in the system he/she can input marks for any student of his course. He/she can also find any student by ID, can update result and he can also see this result by show result. Our system has five modules, they are Insert record, Update record, Display student record, Search record, Delete record these modules and its attributes with entity relationship module presented in figure section.

### 1.2 Purpose:

The project is about to handle all the information management of the student regarding admission and academic information. Also it manages resources which were managed and handled by manpower previously. The main purpose of the project is to integrate distinct sections of the organization into consistent manner so that complex functions can be handled smoothly by any technical or non-technical persons. The project aims at the following matters:

 Automation of admission and enrolment as per board, quota, category and available scats.

Assistance in decision-making.



To manage information of student, courses and result.

Consistently update information of all the students.

The main purpose of the Admin Module is to introduce new things and configure important aspects. For e.g. only admins are authorized to introduced of student addition, modification, searching student, delete information. So the master screens for all these are visible to only admin role. This is done by the Admin Module. Thus the main purpose of the Admin Module is to managing the dynamic working of the system.

### 1.3 Scope:

The scope of the project includes the following:

Any college can use this system as it is not client centric.



Application Support & Maintenance after deployment to production

The Admin Module can be reused for projects as well which have many users with different rights. Hence it is reusable.

### 1.4 Project Overview:

Student Management System (SMS) is a web-based application that tracks current student’s academic information. It maintains academic information for ready access by office staff, students, their faculty advisors, and committee members. Instead of tedious paper work, students will be able to submit required information electronically, and the departments will be able to evaluate the submissions with a much quicker turn around. The Student Management System has been modularized into following modules.

## Chapter 2

### 2.1 System Analysis:

Analysis can be defined as breaking up of any whole so as to find out their nature, function etc. It defines design as to make preliminary sketches of to sketch a pattern or outline for plan. To plan and carry out especially by artistic arrangement or in a skillful wall. System analysis and design can be characterized as a set of techniques and processes, a community of interests, a culture and an intellectual orientation.

### 2.2 Existing System Analysis:

System Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. Here the key question is- what all problems exist in the present system? What must be done to solve the problem? The analysis begins when a user or manager begins a study of the program using the existing system.

During analysis, data collected on the various files, decision points, and transactions handled by the present system. The commonly used tools in the system are Data Flow Diagram, interviews, etc. Training, experience, and common sense are required for the collection of relevant information needed to develop the system. The success of the system depends largely on how clearly the problem is defined, thoroughly investigated, and properly carried out through the choice of solution. A good analysis model should provide not only the mechanisms of problem understanding but also the framework of the solution. Thus it should be studied thoroughly by collecting data about the system. Then the proposed system should be analyzed thoroughly in accordance with the needs. System analysis can be categorized into four parts.

System planning and initial investigation



Information Gathering

Applying analysis tools for structured analysis Feasibility study

Cost/ Benefit analysis.

In the current system we need to keep a number of records related to the student and want to enter the details of the student and the marks manually. In this system only the teacher or the educational institution authority view the mark of the student and they want to enter the details of the student.

### 2.3 Proposed system:

In our proposed system we have the provision for adding the details of the students by themselves. So the overhead of the educational institution authorities and the teachers is become less. Another advantage of the system is that it is very easy to edit the details of the student and delete a student when it found unnecessary. The marks of the student are added in the database and so students can also view the marks whenever they want.

Our proposed system has several advantages

* User friendly interface
* Fast access to database
* Less error
* More Storage Capacity
* Search facility
* Quick transaction

All the manual difficulties in managing the student details in an educational institution have been rectified by implementing computerization.

### 2.4 Feasibility Study:

Feasibility analysis begins once the goals are defined. It starts by generating broad possible solutions, which are possible to give an indication of what the new system should look like. This is where creativity and imagination are used. Analysts must think up new ways of doing things- generate new ideas. There is no need to go into the detailed system operation yet. The solution should provide enough information to make reasonable estimates about project cost and give users an indication of how the new system will fit into the organization. It is important not to exert considerable effort at this stage only to find out that the project is not worthwhile or that there is a need significantly change the original goal. Feasibility of a new system means ensuring that the new system, which we are going to implement, is efficient and affordable. There are various types of feasibility to be determined. They are:

#### 2.4.1 Operation Feasibility:

An estimate should be made to determine how much effort and care will go into the developing of the system including the training to be given to the user. Usually, people are reluctant to changes that come in their progression. The computer initialization will certainly affected the turn over, transfer and employee job status. Hence an additional effort is to be made to train and educate the users on the new way of the system.

#### 2.4.2 Technical Feasibility:

The main consideration is to be given to the study of available resources of the organization where the software is to be implemented. Here the system analyst evaluates the technical merits of the system giving emphasis on the performance, Reliability, maintainability and productivity.

By taking the consideration before developing the proposed system, the resources availability of the organization was studied. The organization was immense computer facilities equipped with sophisticated machines and the software hence this technically feasible.

#### 2.4.3 Economical Feasibility:

Development of this application is highly economically feasible. The only thing to be done is making an environment with an effective supervision. It is cost effective in the sense that has eliminated the paper work completely. The system is also time effective because the calculations are automated which are made at the end of the month or as per the user requirement. Cost benefit analysis is usually performed for this purpose.

## Chapter: 3

### 3.1 Requirement analysis of student management system

**Hardware Requirements of student management system**

**The hardware required for the development of the project is:**



**PROCESSOR:** Intel P-IV system

**PROCESSOR SPEED:** 250MHz to 833MHz

**RAM:** 512MB RAM

**HARD DISK:** 40GB

### 3.2 Software Requirements of student management system

**The software required for the development of the project is:**

**OPERATING SYSTEM:** Windows XP / Vista / 7 / 8.x / 10 / 11



**SOFTWERE :** Code blocks

**VERSION :** 20.03 -32bit

**LANGUAGE:** C language

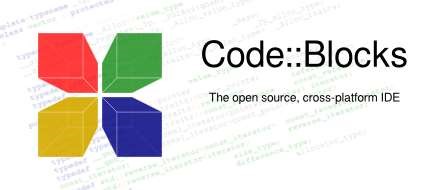
## Chapter 4

### 4.1 Software:

#### 4.1.1 Code::Blocks

Code::Blocks is a free, open-source cross-platform IDE that supports multiple compilers including GCC, Clang and Visual C++. It is developed in C++ using wxWidgets as the GUI toolkit. Using a plugin architecture, its capabilities and features are defined by the provided plugins. Currently, Code::Blocks is oriented towards C, C++, and Fortran. It has a custom build system and optional Make support.

Code::Blocks is being developed for Windows and Linux (the latest macOS version is 13.12 released on 12/26/2013) and has been ported to FreeBSD, OpenBSD and Solaris.



After releasing two release candidate versions, 1.0rc1 on July 25, 2005 and 1.0rc2 on October 25, 2005, instead of making a final release, the project developers started adding many new features, with the final release being repeatedly postponed. Instead, there were nightly builds of the latest SVN version made available on a daily basis.[citation needed] The first stable release was on February 28, 2008, with the version number changed to 8.02. The versioning scheme was changed to that of Ubuntu, with the major and minor number representing the year and month of the release. Version 17.12 is the latest stable release; however for the most up-to-date version the user can download the relatively stable nightly build or download the source code from SVN. Jennic Limited distributes a version of Code::Blocks customized to work with its microcontrollers.[Wikipedia]

### 4.2 Programming Language

#### 4.2.1 C Language

C (/si/, as in the letter c) is a general-purpose, imperative computer programming language, supporting structured programming, lexical variable scope and recursion, while a static type system prevents many unintended operations. By design, C provides constructs that map efficiently to typical machine instructions, and therefore it has found lasting use in applications that had formerly been coded in assembly language, including operating systems, as well as various application software for computers ranging from supercomputers to embedded systems



Many later languages have borrowed directly or indirectly from C, including C, C#, Unix’s C shell, D, Go, Java, JavaScript, Limbo, LPC, Objective-C, Perl, PHP, Python, Rust, Swift, Verilog and SystemVerilog (hardware description languages). These languages have drawn many of their control structures and other basic features

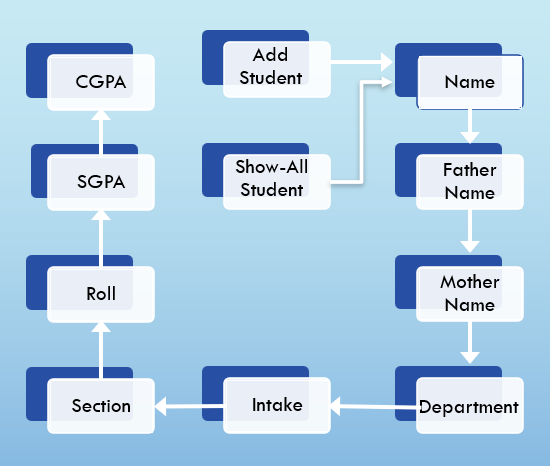
from C.[Wikipedia]

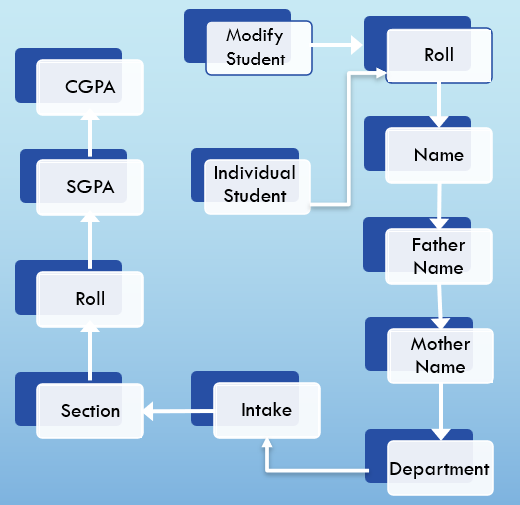
## Chapter 5

### 5.1 Architectural Design

## 5.1.1 Flowchart

## 5.1.2 Sub-Features





## Chapter 6

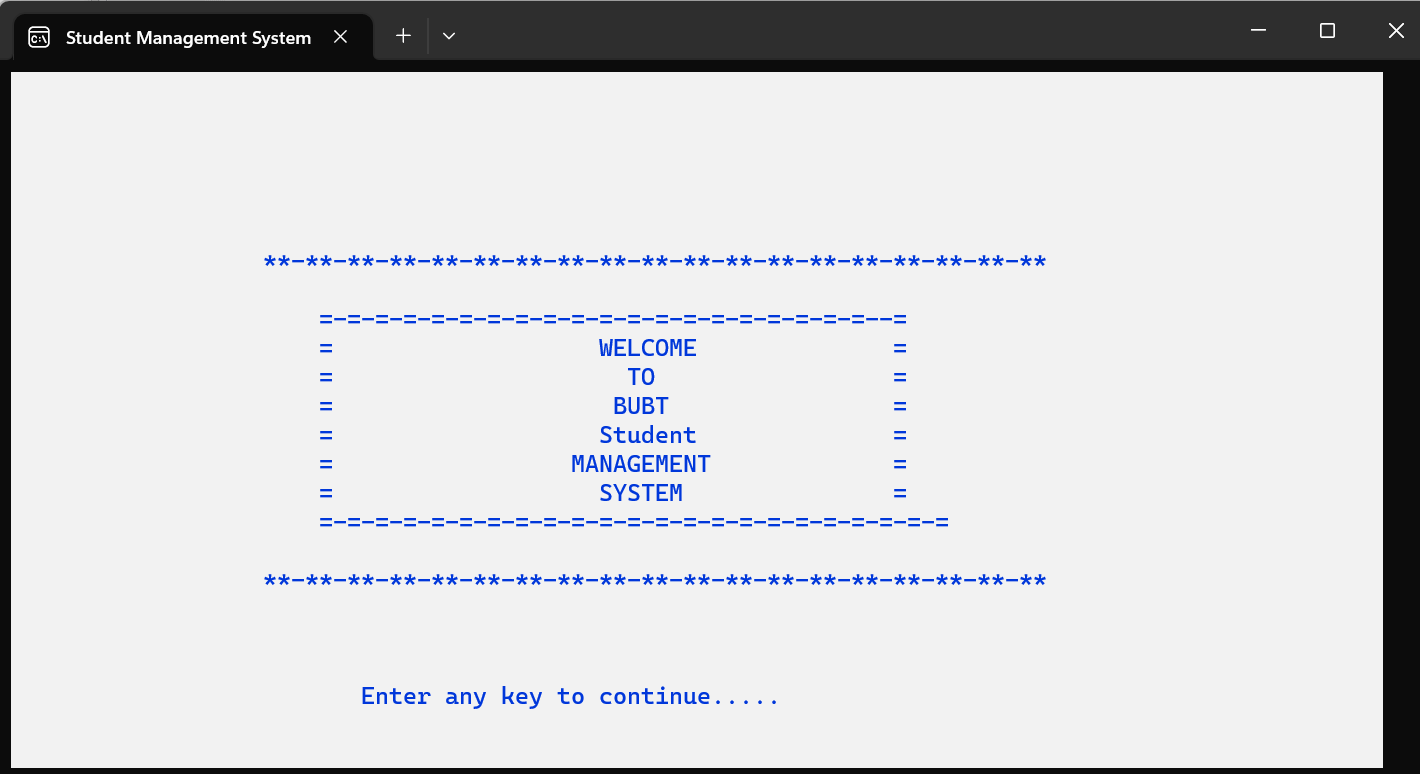
### 6.1 System over View:

Student Management System (SMS) is a software application that tracks current student’s academic information. It maintains academic information for ready access by office, students, their faculty advisors, and committee members. Instead of tedious paper work, students will be able to submit required information electronically, and the departments will be able to evaluate the submissions with a much quicker turn around. **Here all module of this:**

#### 6.1.1. Loading:

The information of the admin of the institute is stored in this entity. It is stored data of login and the password. This provides the security to the system and keeps the record of which user entered in the system at what instance of time. This entity will have the access to all the entity as it will add the student to the system.

#### 6.1.2. Welcome Message:

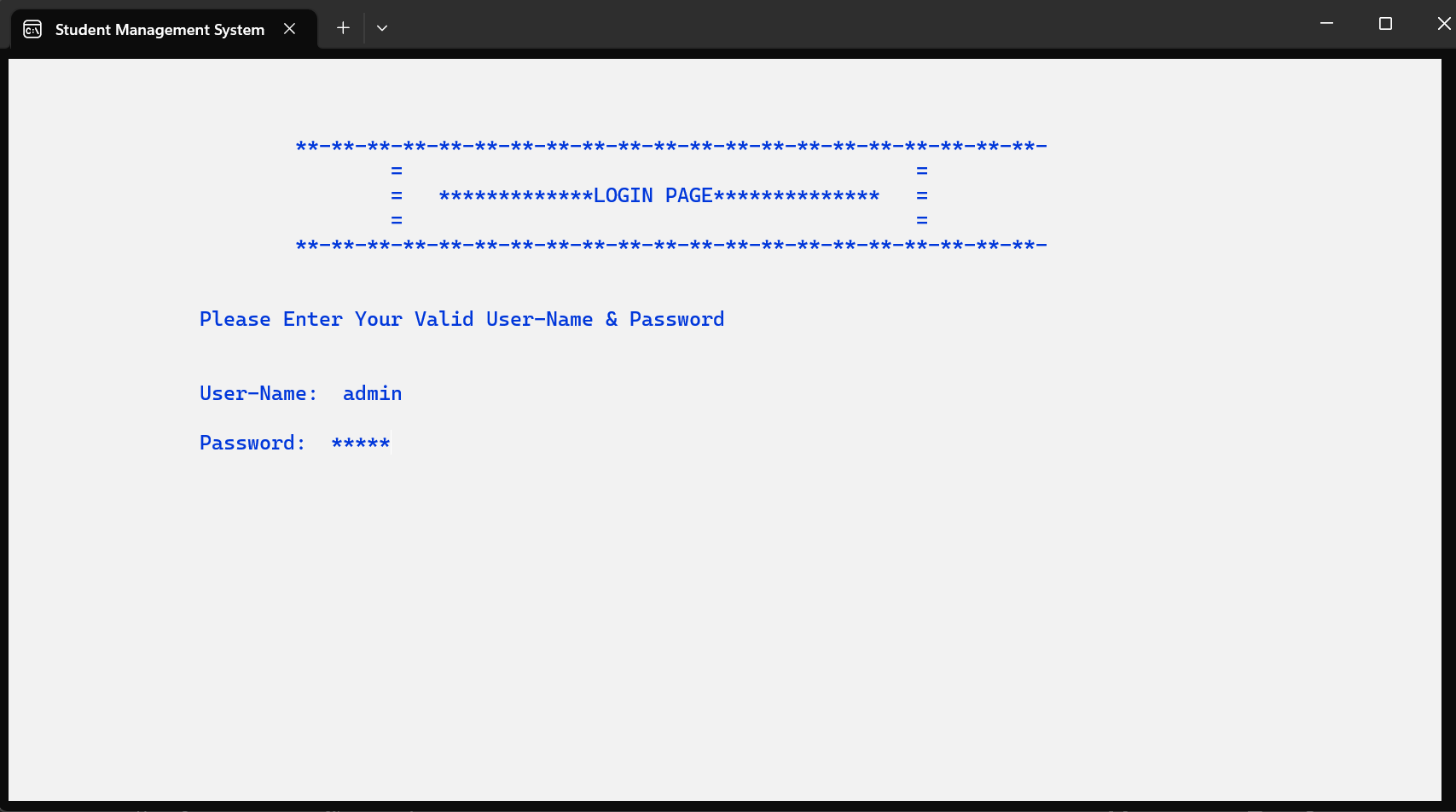


**Fig: 6.1 Welcome message**

Welcome to Student Management Systems Project. This module shows us welcome message.

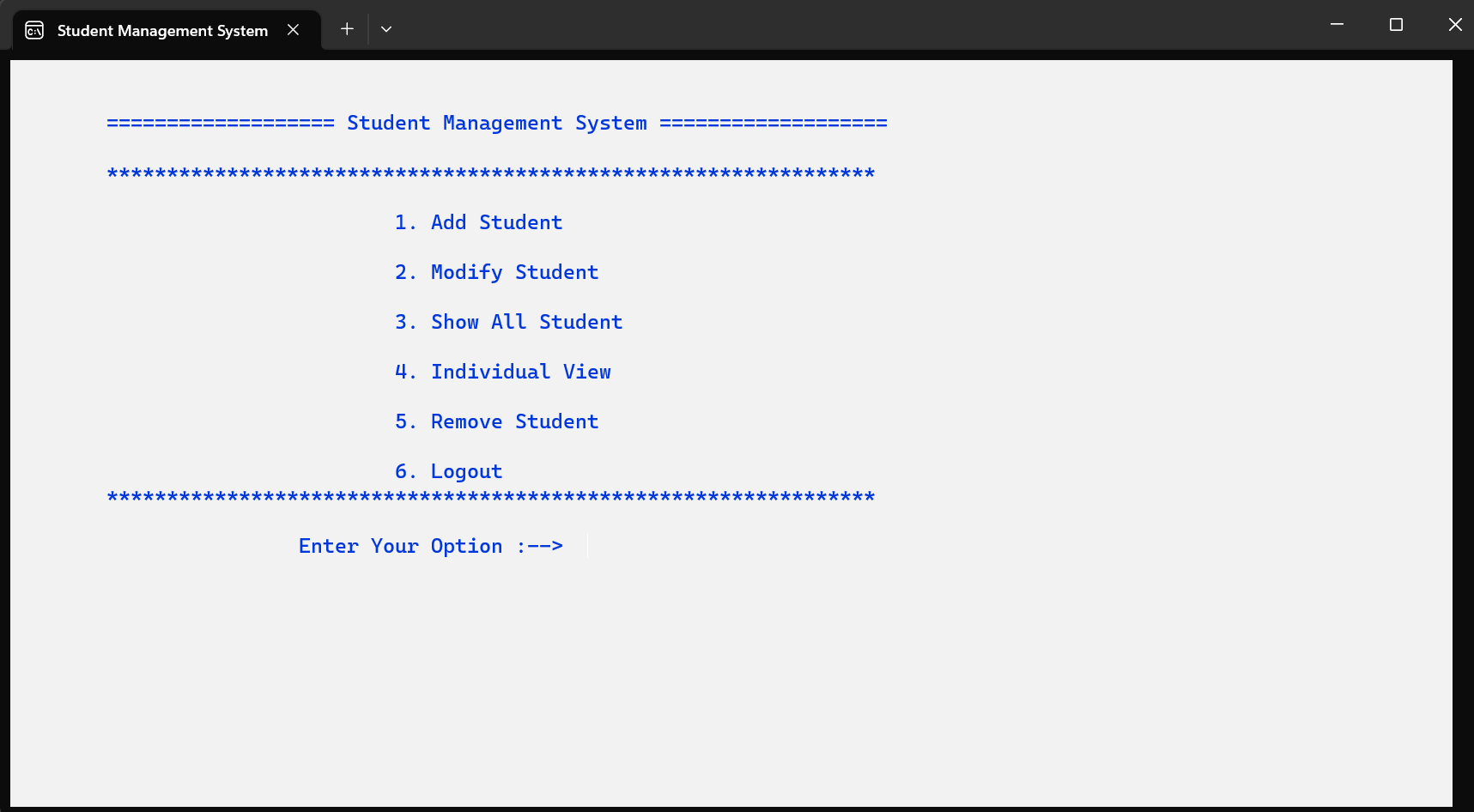
### 6.2 Admin Module:

Each student will be able to select various courses present in the system. It will be published by the administrator under the specific department. Every course has the qualification criteria. It will be available to those students who are eligible for it. All the courses will be handled by the department assigned to them by the administrator.

  **F Fig: 6.2 Admin module**

### 6.3 Menu:

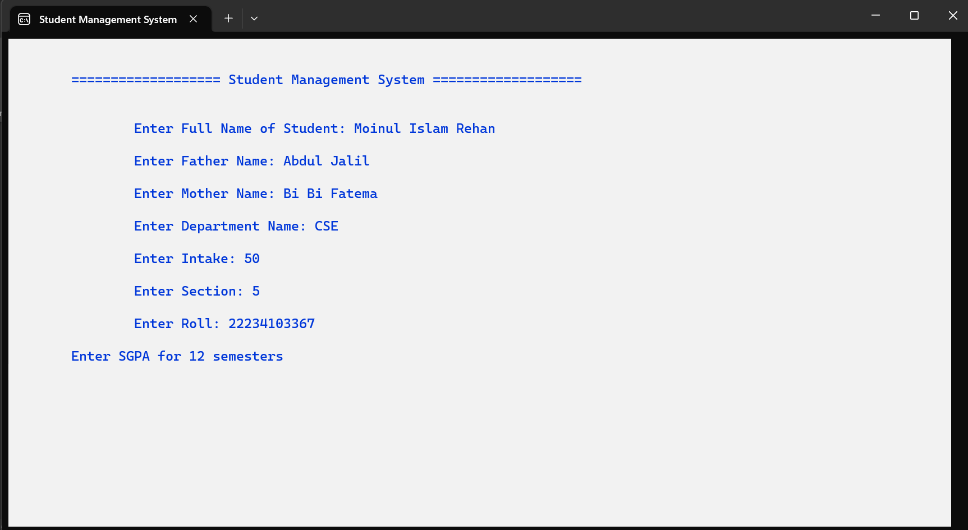
Menu is a function that allows a user of this system to show his or her specific required options.



**Fig: 6.3 Menu**

#### 6.3.1 Add Student:

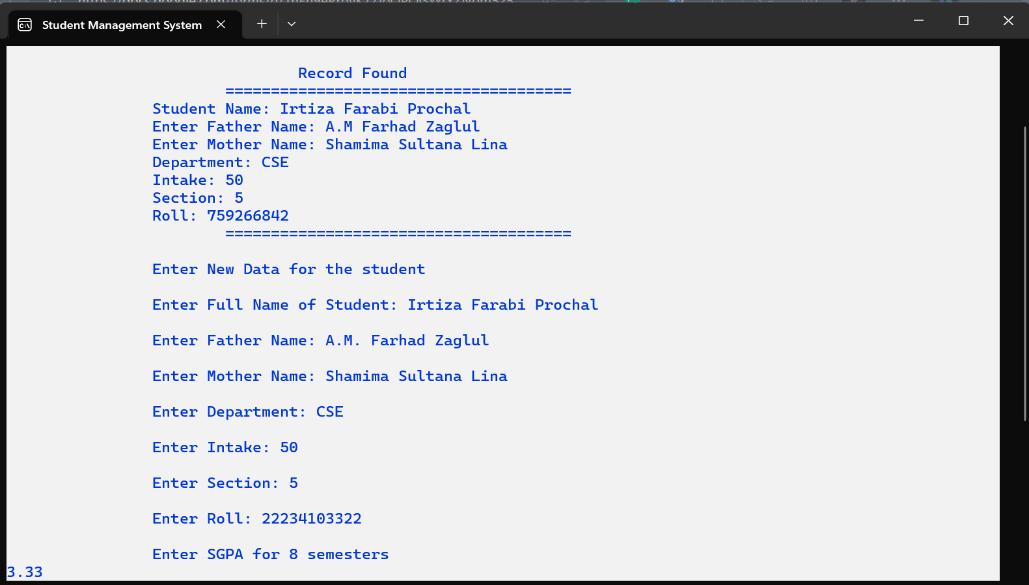
In the institution there will be thousands of students. The only information provided to add is the name of the student.



**Fig: 6.4 Add Student**

#### 6.3.2 Modify Student:

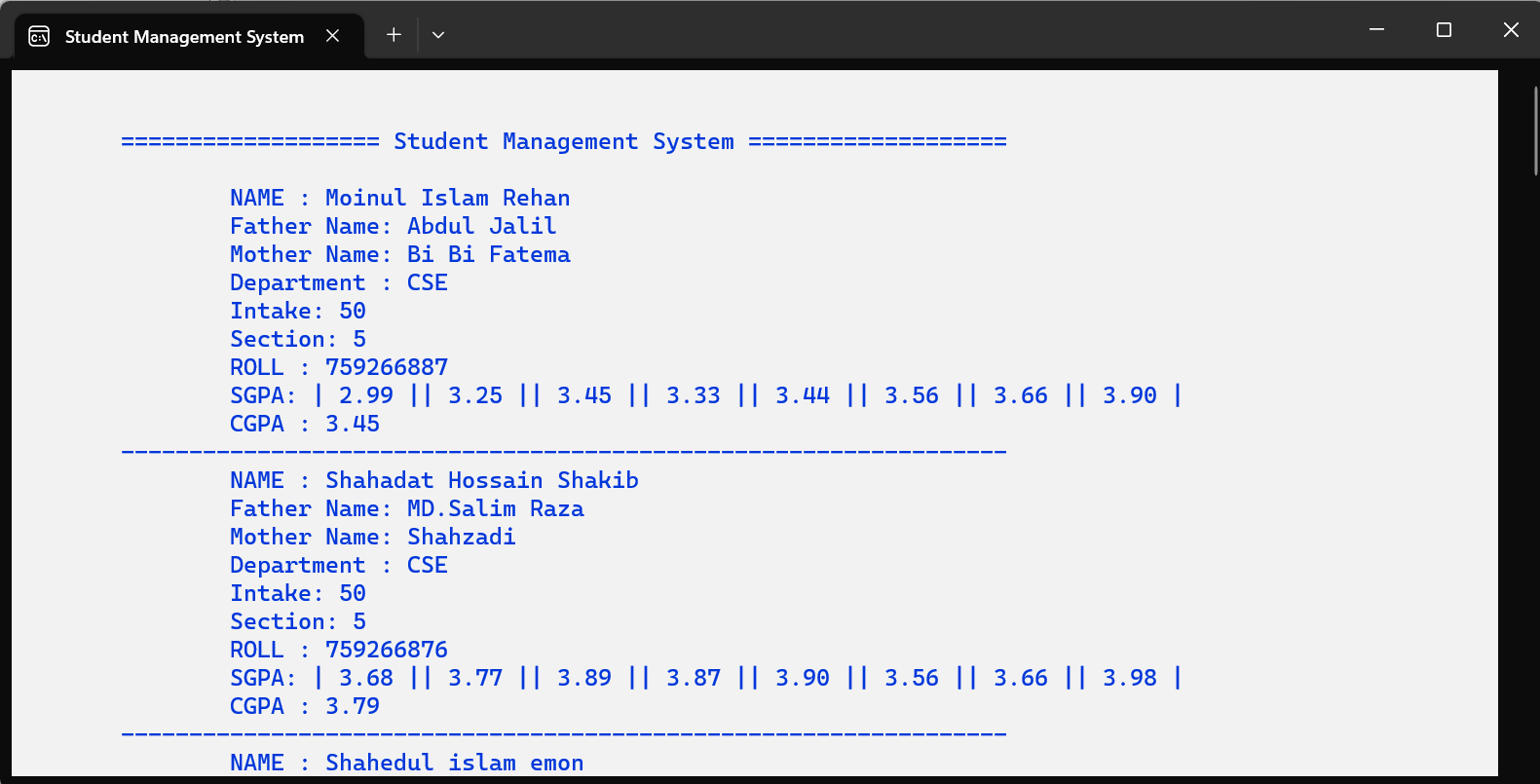
If we want to modify the information of any student during our project, we can add the information of any student through this modify option.



**Fig: 6.5.1 Modify student**

#### 6.3.3 Show all student:

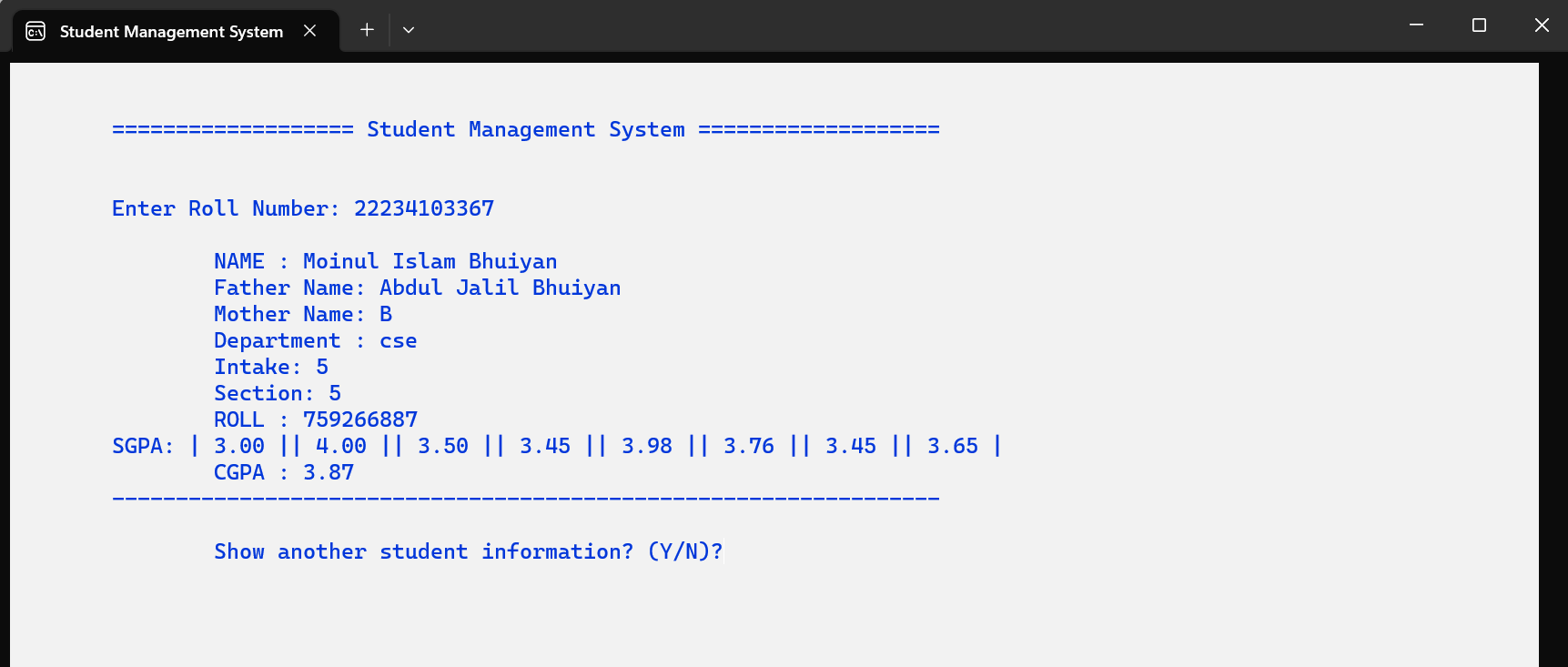
We have added many names to our project. In that case, if we use the show option, we will easily get the information of all the students.



**Fig: 6.6 Show all student**

#### 6.3.4 Individual View:

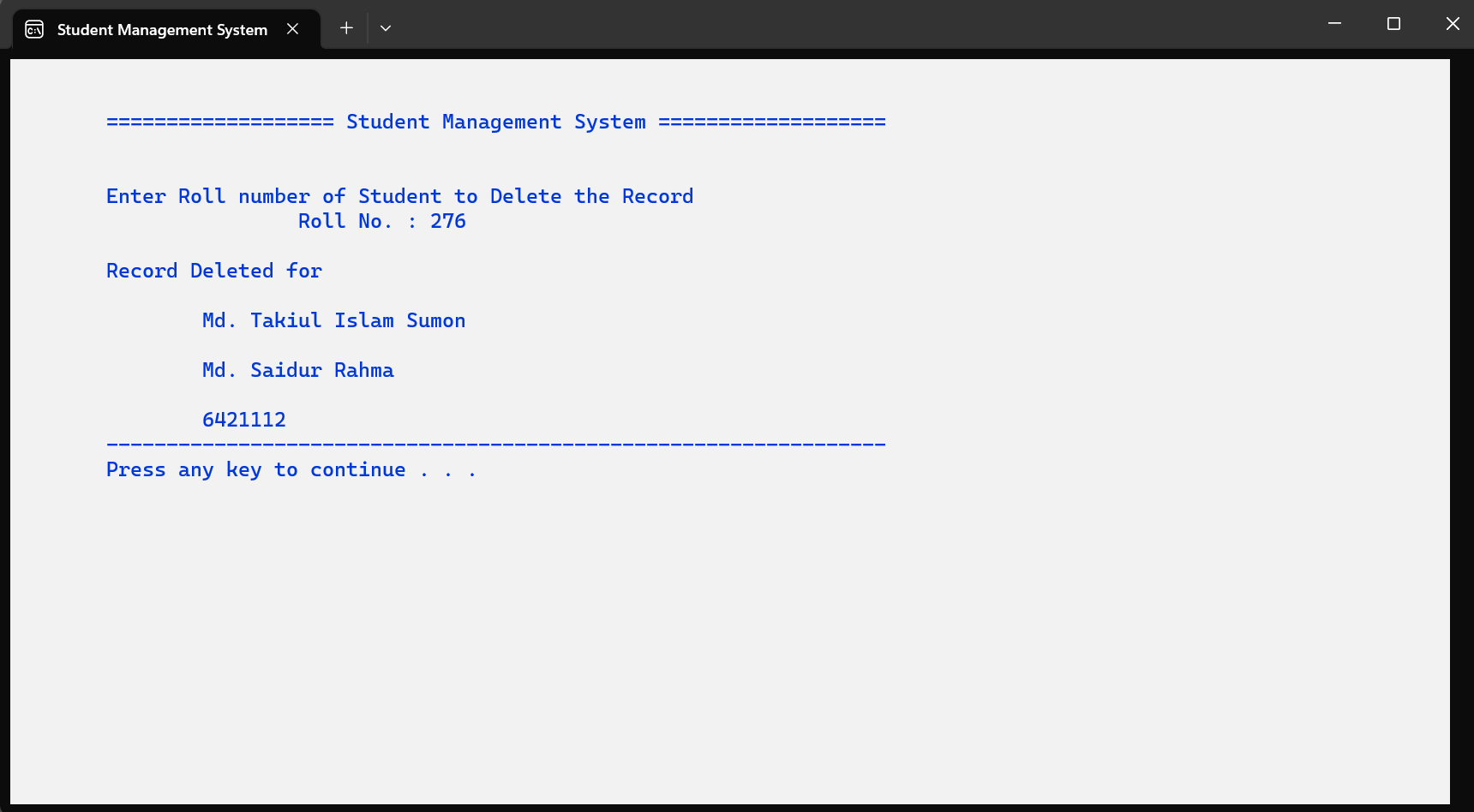
In the institution there will be thousands of students and suppose from this there is need to find the detail of specific student. The only information provided to search is the name of the student. In the manual system, it will be catastrophic to find the student as it is a very tedious job to do so.



**Fig: 6.7 Individual view**

#### 6.3.5 Delete:

If the project has to delete the data of any student of the institution, the data can be deleted effortlessly by this delete function.



**Fig: 6.8 Delete**

## Chapter 7

### 7.1 Code Analysis:

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<math.h>

#include<windows.h>

#define Student struct Stud

void add(FILE \* fp);

void modify(FILE \* fp);

void display(FILE \* fp);

void Indivisual(FILE \*fp);

void password();

FILE \* del(FILE \* fp);

void printChar(char ch,int n);

void title();

FILE \*tp;

void login();

void loading();

void menu();

void welcomeMessage();

//for cursorposition handeling

void gotoxy(int x,int y)

{

COORD CRD;

CRD.X = x;

CRD.Y = y;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE),CRD);

}

COORD coordinates = {0,0};

//for gotoxy handeling

void gotocoordinate(int x,int y)

{

coordinates.X=x;

coordinates.Y=y;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE),coordinates);

}

/\*We are Using setcolor function for Maintaining Text color \*/

void setcolor(int ForgC)

{

WORD wColor;

HANDLE hStdOut=GetStdHandle(STD\_OUTPUT\_HANDLE);

CONSOLE\_SCREEN\_BUFFER\_INFO csbi;

if(GetConsoleScreenBufferInfo(hStdOut,&csbi))

{

wColor=(csbi.wAttributes & 0xF0)+(ForgC & 0x0F);

SetConsoleTextAttribute(hStdOut,wColor);

}

}

struct pass

{

char pass[25];

} pa;

//for structure

struct Stud

{

char name[100];

char fathername[100];

char mothername[100];

char dept[50];

int intake;

char sec;

int roll;

float sgpa[8];

float cgpa;

};

//Fast loading page

void loading()

{

SetConsoleTitle("Student Management System || BUBT");

int r,q;

gotoxy(40,10);

setcolor(1);

printf("LOADING......");

gotoxy(36,12);

for(r=1; r<=20; r++)

{

for(q=0; q<=100000000; q++);

printf("%c",177);

}

system("cls");

}

//This is welcome message

void welcomeMessage()

{

SetConsoleTitle("Student Management System || BUBT");

system("cls");

setcolor(1);

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

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

printf("\n\t\t =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=--=");

printf("\n\t\t = WELCOME =");

printf("\n\t\t = TO =");

printf("\n\t\t = BUBT =");

printf("\n\t\t = Student =");

printf("\n\t\t = MANAGEMENT =");

printf("\n\t\t = SYSTEM =");

printf("\n\t\t =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=");

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

printf("\n\n\n\t\t\t Enter any key to continue.....");

getch();

}

//For loging function

void login()

{

system("cls");

int valid=0;

int n,p;

int i=0;

char username[20],a;

char password[10];

//gotoxy(20,8);

SetConsoleTitle("Student Management System || BUBT");

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

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

printf("\n\t\t\t\t= \*\*\*\*\*\*\*\*\*\*\*\*\*LOGIN PAGE\*\*\*\*\*\*\*\*\*\*\*\*\*\* =");

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

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

printf("\n\n\t\tPlease Enter Your Valid User-Name & Password\n");

printf("\n\n\t\tUser-Name: ");

scanf("%s",username);

printf("\n\t\tPassword: ");

while(1)

{

a = getch();

if(a ==13)

{

break;

}

printf("\*");

password[i] = a;

i++;

}

password[i]='\0';

n = strcmp("admin",username);

p = strcmp("12345",password);

if(n == 0 && p == 0)

{

menu();

}

else

{

printf("\n\n\t!!!! ERROR !!!!Wrong Password Or User Name Please Enter Valid Password And User Name");

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

system("pause");

login();

}

}

//Main funtion

int main()

{

system("COLOR F2");

loading();

welcomeMessage();

login();

menu();

}

// Menu function

void menu()

{

system("COLOR F1");

int ch,id,k,i;

char c,pas[50];

SetConsoleTitle("Student Management System || BUBT");

FILE \* fp;

Student s;

int option;

char another;

if((fp=fopen("information.txt","rb+"))==NULL)

{

if((fp=fopen("information.txt","wb+"))==NULL)

{

printf("Can't create or open Database.");

return 0;

}

}

while(1)

{

title();

printf("\n\t");

printChar('\*',64);

printf("\n\n\t\t\t\t1. Add Student");

printf("\n\n\t\t\t\t2. Modify Student");

printf("\n\n\t\t\t\t3. Show All Student");

printf("\n\n\t\t\t\t4. Individual View");

printf("\n\n\t\t\t\t5. Remove Student");

printf("\n\n\t\t\t\t6. Logout\n\t");

printChar('\*',64);

printf("\n\n\t\t\tEnter Your Option :--> ");

scanf("%d",&option);

switch(option)

{

case 1:

add(fp);

break;

case 2:

modify(fp);

break;

case 3:

display(fp);

break;

case 4:

Indivisual(fp);

break;

case 5:

fp=del(fp);

break;

case 6:

exit(0);

default:

printf("\n\t\tNo Action Detected");

printf("\n\t\tPress Any Key........\n");

getch();

system("pause");

}

}

return 1;

}

void printChar(char ch,int n)

{

while(n--)

{

putchar(ch);

}

}

void title()

{

system("cls");

system("COLOR F1");

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

printChar('=',19);

printf(" Student Management System ");

printChar('=',19);

printf("\n");

}

//Insert at end

//case 1

void add(FILE \*fp)

{

title();

char another='y';

Student s;

int i;

float cgpa;

fseek(fp,0,SEEK\_END);

while(another=='y'||another=='Y')

{

printf("\n\n\t\tEnter Full Name of Student: ");

fflush(stdin);

fgets(s.name,100,stdin);

s.name[strlen(s.name)-1]='\0';

printf("\n\t\tEnter Father Name: ");

fflush(stdin);

fgets(s.fathername,100,stdin);

s.fathername[strlen(s.fathername)-1]='\0';

printf("\n\t\tEnter Mother Name: ");

fflush(stdin);

fgets(s.mothername,100,stdin);

s.mothername[strlen(s.mothername)-1]='\0';

printf("\n\t\tEnter Department Name: ");

fflush(stdin);

fgets(s.dept,50,stdin);

s.dept[strlen(s.dept)-1]='\0';

printf("\n\t\tEnter Intake: ");

scanf("%d",&s.intake);

printf("\n\t\tEnter Section: ");

scanf("%d",&s.sec);

printf("\n\t\tEnter Roll: ");

scanf("%d",&s.roll);

printf("\n\tEnter SGPA for 8 semesters\n");

for(i=0,cgpa=0; i<8; i++)

{

scanf("%f",&s.sgpa[i]);

cgpa+=s.sgpa[i];

}

cgpa/=8.0;

s.cgpa=cgpa;

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

printf("\n\n\t\tAdd another student?(Y/N)?");

fflush(stdin);

another=getchar();

}

}

//Delete function cas3 3

FILE \* del(FILE \* fp)

{

title();

Student s;

int flag=0,tempRoll,siz=sizeof(s);

FILE \*ft;

if((ft=fopen("temp.txt","wb+"))==NULL)

{

printf("\n\n\t\t\t\\t!!! ERROR !!!\n\t\t");

system("pause");

return fp;

}

printf("\n\n\tEnter Roll number of Student to Delete the Record");

printf("\n\t\t\tRoll No. : ");

scanf("%d",&tempRoll);

rewind(fp);

while((fread(&s,siz,1,fp))==1)

{

if(s.roll==tempRoll)

{

flag=1;

printf("\n\tRecord Deleted for");

printf("\n\n\t\t%s\n\n\t\t%s\n\n\t\t%d\n\t",s.name,s.fathername,s.mothername,s.dept,s.intake,s.sec,s.roll);

continue;

}

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

}

fclose(fp);

fclose(ft);

remove("information.txt");

rename("temp.txt","information.txt");

if((fp=fopen("information.txt","rb+"))==NULL)

{

printf("ERROR");

return NULL;

}

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

printChar('-',65);

printf("\n\t");

system("pause");

return fp;

}

// case 2 Modify

void modify(FILE \*fp)

{

title();

Student s;

int i,flag=0,tempRoll,siz=sizeof(s);

float cgpa;

printf("\n\n\tEnter Roll Number of Student to MODIFY the Record : ");

scanf("%d",&tempRoll);

rewind(fp);

while((fread(&s,siz,1,fp))==1)

{

if(s.roll==tempRoll)

{

flag=1;

break;

}

}

if(flag==1)

{

fseek(fp,-siz,SEEK\_CUR);

printf("\n\n\t\t\t\tRecord Found\n\t\t\t");

printChar('=',38);

printf("\n\t\tStudent Name: %s",s.name);

printf("\n\t\tEnter Father Name: %s",s.fathername);

printf("\n\t\tEnter Mother Name: %s",s.mothername);

printf("\n\t\tDepartment: %s",s.dept);

printf("\n\t\tIntake: %d",s.intake);

printf("\n\t\tSection: %d",s.sec);

printf("\n\t\tRoll: %d\n\t\t\t",s.roll);

printChar('=',38);

printf("\n\n\t\tEnter New Data for the student");

printf("\n\n\t\tEnter Full Name of Student: ");

fflush(stdin);

fgets(s.name,100,stdin);

s.name[strlen(s.name)-1]='\0';

printf("\n\t\tEnter Father Name: ");

fflush(stdin);

fgets(s.fathername,100,stdin);

s.fathername[strlen(s.fathername)-1]='\0';

printf("\n\t\tEnter Mother Name: ");

fflush(stdin);

fgets(s.mothername,100,stdin);

s.mothername[strlen(s.mothername)-1]='\0';

printf("\n\t\tEnter Department: ");

fflush(stdin);

fgets(s.dept,50,stdin);

s.dept[strlen(s.dept)-1]='\0';

printf("\n\t\tEnter Intake: ");

scanf("%d",&s.intake);

printf("\n\t\tEnter Section: ");

scanf("%d",&s.intake);

printf("\n\t\tEnter Roll: ");

scanf("%d",&s.roll);

printf("\n\t\tEnter SGPA for 8 semesters\n");

for(i=0,cgpa=0; i<8; i++)

{

scanf("\t%f",&s.sgpa[i]);

cgpa+=s.sgpa[i];

}

cgpa=cgpa/8.0;

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

}

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

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

system("pause");

}

//For Display show

void display(FILE \* fp)

{

title();

Student s;

int i,siz=sizeof(s);

rewind(fp);

while((fread(&s,siz,1,fp))==1)

{

printf("\n\t\tNAME : %s",s.name);

printf("\n\t\tFather Name: %s",s.fathername);

printf("\n\t\tMother Name: %s",s.mothername);

printf("\n\t\tDepartment : %s",s.dept);

printf("\n\t\tIntake: %d",s.intake);

printf("\n\t\tSection: %d",s.sec);

printf("\n\t\tROLL : %d",s.roll);

printf("\n\t\tSGPA: ");

for(i=0; i<8; i++)

printf("| %.2f |",s.sgpa[i]);

printf("\n\t\tCGPA : %.2f\n\t",s.cgpa);

printChar('-',65);

}

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

printChar('\*',65);

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

system("pause");

}

//For indivisual case 4

void Indivisual(FILE \*fp)

{

title();

int tempRoll,flag,siz,i;

Student s;

char another='y';

siz=sizeof(s);

while(another=='y'||another=='Y')

{

printf("\n\n\tEnter Roll Number: ");

scanf("%d",&tempRoll);

rewind(fp);

while((fread(&s,siz,1,fp))==1)

{

if(s.roll==tempRoll)

{

flag=1;

break;

}

}

if(flag==1)

{

printf("\n\t\tNAME : %s",s.name);

printf("\n\t\tFather Name: %s",s.fathername);

printf("\n\t\tMother Name: %s",s.mothername);

printf("\n\t\tDepartment : %s",s.dept);

printf("\n\t\tIntake: %d",s.intake);

printf("\n\t\tSection: %d",s.sec);

printf("\n\t\tROLL : %d",s.roll);

printf("\n\tSGPA: ");

for(i=0; i<8; i++)

printf("| %.2f |",s.sgpa[i]);

printf("\n\t\tCGPA : %.2f\n\t",s.cgpa);

printChar('-',65);

}

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

printf("\n\n\t\tShow another student information? (Y/N)?");

fflush(stdin);

another=getchar();

}

}

## Chapter 8

### 8.1 REFRENCES

# 

[https://code-projects.org](https://code-projects.org/) <https://www.wikipedia.org/><https://www.google.com/><https://www.geeksforgeeks.org/><https://www.lovelycoding.org/student-management-system/><http://edujournal.in/><http://freesourcecode.net/><https://www.w3schools.com/><https://www.youtube.com/>

## Chapter 9

# *9.1 Future Plan:*

In future, we plan to add all academic information to it, add an online library and add all kinds of routine facilities. We will not limit it only to BUBT so that all universities and educational institutions can use it easily, we will add user-friendly modifications and features. And for users' convenience, we are planning to bring it as an app as well. And we hope all users can use it comfortably.

## Chapter 10

# *10.1 Conclusion:*

This application is prepared for the established **Bangladesh University of Business and Technology (BUBT)** and it will provide almost full information about how to manage the information of the student. The both utilities provided make the application attractive and easy in getting information. After implementing the application It will contain the advantages were incomparable to the present contemporary systems used by company. The most admirable feature founded was its simplicity in terms of application to the user but its highly beneficial outputs can’t be ignored. The users will be highly benefited after using the system.

**THANK YOU**