

*Bangladesh University of Business and Technology
(BUBT)*



Project Book

Project Name: Student Management System

Course Title : Software Development 1

Course Code : CSE 100

Submitted to:

Umme Hafcha Mukti

Course Instructor

Department of CSE

Bangladesh University of Business and Technology (BUBT)

Team Member :

<i>Name</i>	<i>ID</i>	<i>Intake</i>	<i>Section</i>
<i>Anika Tahsin Prokriti</i>	<i>19202103457</i>	<i>44</i>	<i>12</i>
<i>MD. Shariar Rahman Oion</i>	<i>19202103458</i>	<i>44</i>	<i>12</i>
<i>Sayed Samiul Hoque</i>	<i>19202103465</i>	<i>44</i>	<i>12</i>

“GOD HELPS THOSE WHO HELP THEMSELVES”

MAY ALLAH HELP US FOR SUCCESS

(AMEEN)

Table of Contents

Dedication	6
Abstract	7
Acknowledgement	8
Declaration	9
Copyright	10
Certificate	11
Approval	12
Chapter 1	13
1.1 Introduction	13
1.2 Purpose:	13
1.3 Scope:	14
1.4 Project Overview:	14
Chapter 2	15
2.1 System Analysis:	15
2.2 Existing System Analysis:	15
2.3 Proposed system:	16
2.4 Feasibility Study:	16
2.4.1 Operation Feasibility:	17
2.4.2 Technical Feasibility:	17
2.4.3 Economical Feasibility:	17
Chapter: 3	18
3.1 Requirement analysis of student management system	18
3.2 Software Requirements of student management system	18
Chapter 4	19
4.1 Software:	19
4.1.1 Code::Blocks	19
4.2 Programming Language	20
4.2.1 C Language	20
Chapter 5	21
5.1 File Handling	21

5.2 File Handling in C	21
5.3 Different Types of Files in C	21
5.3.1 Text files	22
5.3.2 Binary files	22
5.4 C File Handling Operations	22
5.5 Opening a Text File in C	23
5.6 Closing a Text File in C	24
5.7 Reading and Writing a Text File in C	24
5.8 Reading and Writing a Binary File in C	24
Chapter 6	25
6.1 System over View:	25
6.1.1. Loading:	25
6.1.2. Welcome Message:	25
6.2 Admin Module:	26
6.3 Menu:	26
6.3.1 Add Student:	27
6.3.2 Modify Student:	27
6.3.3 Show all student:	28
6.3.4 Individual View:	29
6.3.5 Delete:	29
Chapter 7	30
7.1 Coding Optimization:	30
Chapter 8	31
Testing	31
8.1 Introduction of Testing	31
8.2 Testing and Debugging:	31
8.3 Function Testing	32
8.4 Structural Testing:	32
Chapter 9	36
9.1 Code Analysis:	36
Chapter 10	57
10.1 REFERENCES	57
Website References:	57
Chapter 11	58

11.1 Conclusion: -----58



Dedication

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

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 project Instructor **Umme Hafcha Mukti** for her 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 her. 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,

Anika Tahsin Prokriti: - 19202103457

MD.Shariar Rahman Oion:-19202103458

Sayed Samiul Hoque: - 19202103465

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.

Anika Tahsin Prokriti

ID(19202103457)

MD. Shariar Rahman Oion

ID(19202103458)

Sayed Samiul Hoque

ID(19202103465)

Copyright

© *Copyright by*

Anika Tahsin Prokriti, MD. Shariar Rahman Oion & Sayed Samiul Hoque

2021 All Rights Reserved

Certificate

This is to certify that by **Anika Tahsin Prokriti(ID-19202103457)**, **MD. Shariar Rahman Oion(ID-19202103458)**, & **Sayed Samiul Hoque (ID-19202103465)**, were belong to the department of Computer Science and Engineering, have completed their project work titled **Student Management System** satisfactorily in partial fulfillment for the requirement of *Bachelor of Science in Computer Science and Engineering* of **Bangladesh University of Business and Technology(BUBT)** in the year 2021.

Course Instructor

Umme Hafcha Mukti

Department of Computer Science and Engineering

Bangladesh University of Business and Technology (BUBT)

Approval

The project work entitled **Student Management System** is submitted by **Anika Tahsin Prokriti (ID-19202103457)**, **MD. Shariar Rahman Oion (ID-19202103458)**, & **Syed Samiul Haque (ID-19202103465)** 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.

Course Instructor

Umme Hafcha Mukti

Department of Computer Science and Engineering

Bangladesh University of Business and Technology (BUBT)

Chairman

Dr. Muhammad Firoz Mridha

Department of Computer Science and Engineering

Bangladesh University of Business and Technology (BUBT)





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 seats.
-  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. It is the comparative study of the cost verses the benefit and savings that are expected from the proposed system. Since the organization is well equipped with the required hard ware, the project was found to be economically

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
-  **SOFTWARE :** Code blocks
-  **VERSION :** 20.03 -32bit
-  **LANGUAGE:** C language

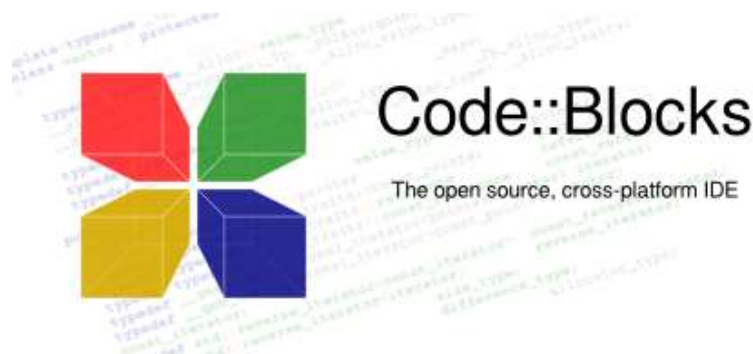
Chapter 4

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 File Handling

File handling in C refers to the task of storing data in the form of input or output produced by running C programs in data files, namely, a text file or a binary file for future reference and analysis. Once we compile and run the program, the output is obtained, but this output is not stored in the form of information anywhere in the system. This problem can easily be solved by the implementation of file handling in C. Since most of the computer systems work with files as it helps in storing information, C offers this benefit of file handling.

5.2 File Handling in C

The contents of a file are not volatile like the C compiler memory. The various operations available like creating a file, opening a file, reading a file or manipulating data inside a file is referred to as file handling. Here are some of the following reasons behind the popularity of file handling:

- ✚ **Reusability:** It helps in preserving the data or information generated after running the program.
- ✚ **Large storage capacity:** Using files, you need not worry about the problem of storing data in bulk.
- ✚ **Saves time:** There are certain programs that require a lot of input from the user. You can easily access any part of the code with the help of certain commands.
- ✚ **Portability:** You can easily transfer the contents of a file from one computer system to another without having to worry about the loss of data.

5.3 Different Types of Files in C

There are basically 2 distinct types of data files available in the C programming language:

5.3.1 Text files

These are the simplest files a user can create when dealing with file handling in C. It is created with a **.txt** extension using any simple text editor. Generally, we use notepads to create text files. Hence, it is safe to say that text files are simple to use and access. But, along with advantages comes disadvantages as well. Since it is easily readable, it does not provide any security of information. Moreover, it consumes large storage space. Hence, there is a different type of file available called binary files which helps us solve this problem.

5.3.2 Binary files

A binary file stores information in the form of the binary number system (0's and 1's) and hence occupies less storage space. In simple words, it stores information in the same way as the information is held in computer memory. Therefore, it proves to be much easier to access. It is created with a **.bin** extension. It overcomes the drawback offered by text files. Since it is not readable to humans, the information is more secure. Hence, it is safe to say that binary files prove to be the best way to store information in a data file.

5.4 C File Handling Operations

The C programming offers various operations associated with file handling. They are:

- + Creating a new file: **fopen()**
- + Opening an existing file in your system: **fopen()**
- + Closing a file: **fclose()**
- + Reading characters from a line: **getc()**
- + Writing characters in a file: **putc()**
- + Reading a set of data from a file: **fscanf()**
- + Writing a set of data in a file: **fprintf()**
- + Reading an integral value from a file: **getw()**
- + Writing an integral value in a file: **putw()**
- + Setting a desired position in the file: **fseek()**
- + Getting the current position in the file: **ftell()**
- + Setting the position at the beginning point: **rewind()**

It is important to note that a file-type pointer needs to be declared when working with files. It establishes communication between the file and the program.

FILE *fpointer;

Out of these various operations, there are some basic operations used in the [C](#) programming language which we will discuss in detail one by one:

5.5 Opening a Text File in C

We use the `fopen()` function to create or open a file as mentioned earlier. It is pretty obvious that creating or opening a file is the first step in file handling. Once the file has been created, it can be opened, modified or deleted.

The following modes in which the file can be opened are:




MODE	ELUCIDATION
r	We use it to open a text file in reading mode
w	We use it to open or create a text file in writing mode
a	We use it to open a text file in append mode
r+	We use it to open a text file in both reading and writing mode
w+	We use it to open a text file in both reading and writing mode
a+	We use it to open a text file in both reading and writing mode
rb	We use it to open a binary file in reading mode
wb	We use it to open or create a binary file in writing mode
ab	We use it to open a binary file in append mode
rb+	We use it to open a binary file in both reading and writing mode
wb+	We use it to open a binary file in both reading and writing mode
ab+	We use it to open a binary file in both reading and writing mode

5.6 Closing a Text File in C

We use the **fclose()** function to close a file that is already open. It is pretty obvious that the file needs to be opened so that the operation to close a file can be performed.

5.7 Reading and Writing a Text File in C

It is important to note that there are 3 types of streams (sequence of bytes) in a file:

-  Input
-  Output
-  Input / Output

The input/output operations in a file help you read and write in a file. The simplest functions used while performing operations on reading and writing characters in a file are `getc()` and `putc()` respectively. In order to read and write a set of data in a file, we use the `fscanf()` and `fprintf()` operators.

5.8 Reading and Writing a Binary File in C

We use the `fread()` and `fwrite()` function to read and write data from a binary file respectively. Working with binary files is a bit more complicated than working with text files as the syntax involves the use of more arguments. The function `fread()` and `fwrite()` both take 4 arguments to work.

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:

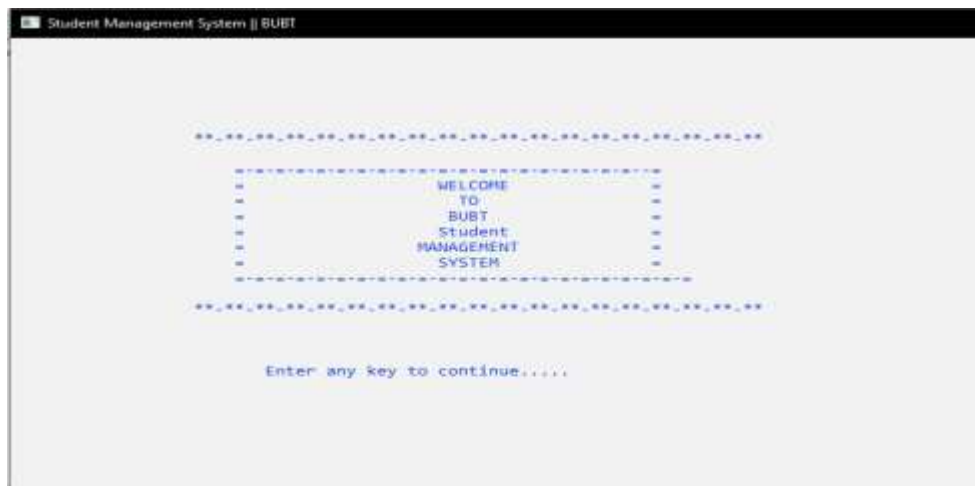


Fig: 6.1 Welcome message

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.



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.

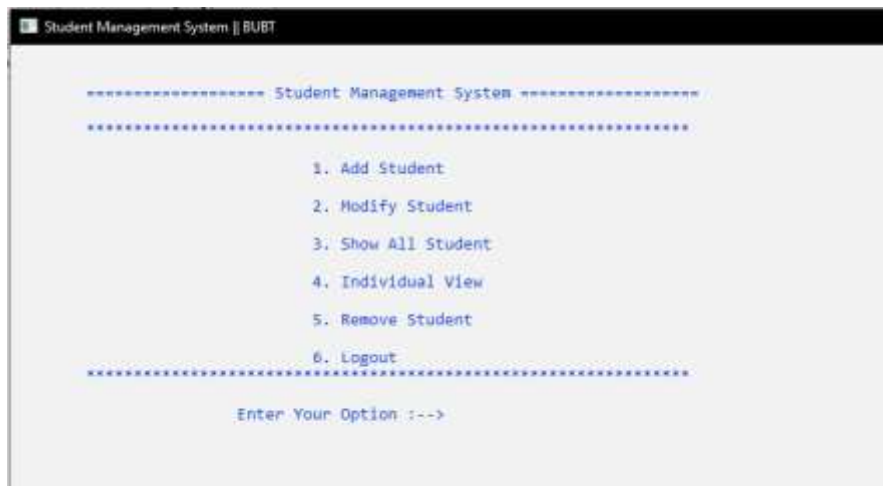
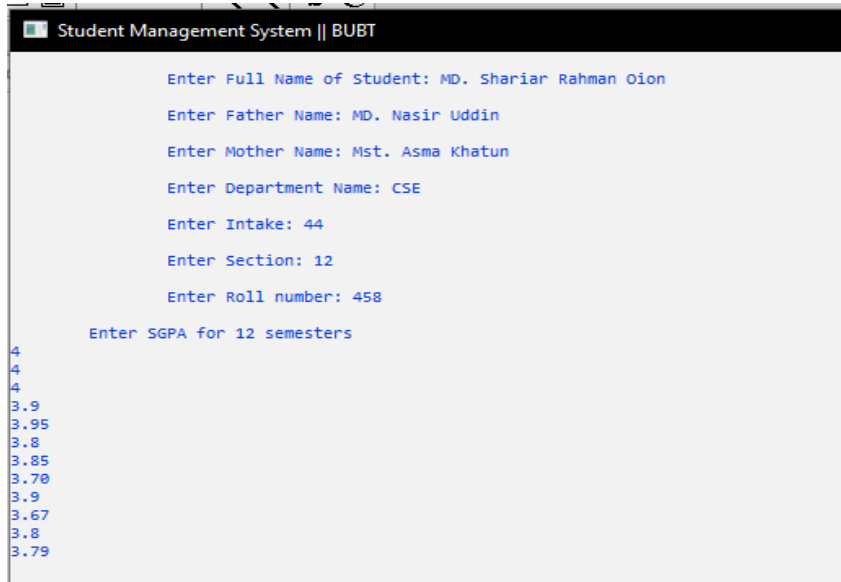


Fig: 6.3 Menu

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.



The screenshot shows a terminal window titled "Student Management System || BUBT". It displays a series of prompts for adding a student's information:

```

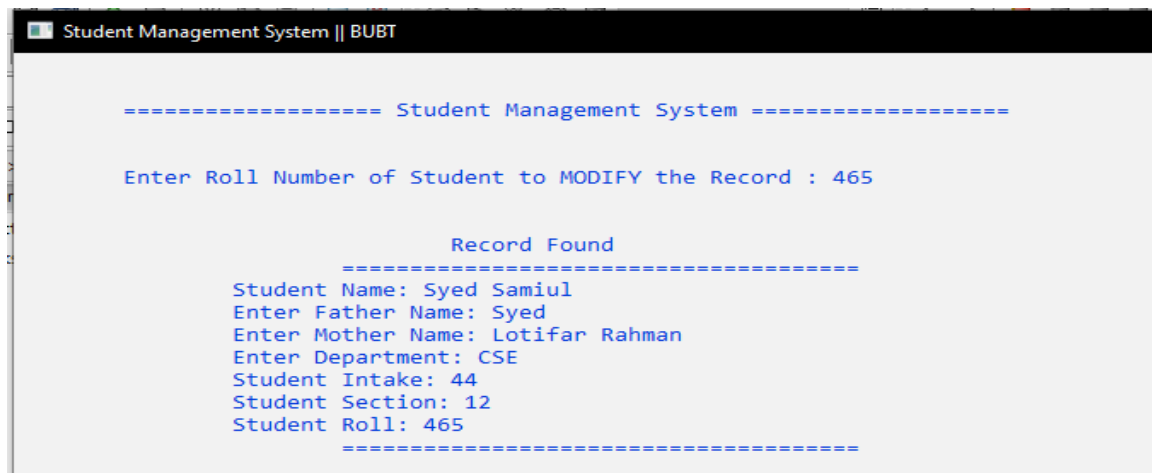
Enter Full Name of Student: MD. Shariar Rahman Oion
Enter Father Name: MD. Nasir Uddin
Enter Mother Name: Mst. Asma Khatun
Enter Department Name: CSE
Enter Intake: 44
Enter Section: 12
Enter Roll number: 458
Enter SGPA for 12 semesters
4
4
4
3.9
3.95
3.8
3.85
3.78
3.9
3.67
3.8
3.79

```

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.



The screenshot shows a terminal window titled "Student Management System || BUBT". It displays the following prompts and output for modifying a student's record:

```

===== Student Management System =====

Enter Roll Number of Student to MODIFY the Record : 465

Record Found
=====
Student Name: Syed Samiul
Enter Father Name: Syed
Enter Mother Name: Lotifar Rahman
Enter Department: CSE
Student Intake: 44
Student Section: 12
Student Roll: 465
=====

```

Fig: 6.5 Modify student

```

Student Management System || BUBT
Student Roll: 465
-----
Enter New Data for the student
Enter Full Name of Student: Syed Samiul Houque
Enter Father Name: Syed Pahalobi Houque
Enter Mother Name: Lotifar Rahman
Enter Department: CSE
Enter Intake: 44
Enter Section: 12
Enter Roll number: 465
Enter SGPA for 12 semesters
4
4
4
3.95
3.8
3.6
4
3.78
3.85
3.99

```

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.

```

Student Management System || BUBT
===== Student Management System =====
NAME : Anika Tahsin
Father Name: Abdus Samad
Mother Name: Bilara Yasmin
Department : CSE
Enter Department: CSE
Student Intake: 44
Student Section: 12
ROLL : 457
SGPA: | 4.00 || 4.00 || 4.00 || 4.00 || 3.70 || 3.80 || 3.90 || 3.97 || 4.00 || 4.00 || 3.30 || 4.00 |
CGPA : 3.89
-----
NAME : MD. Shariar Rahman Oion
Father Name: MD. Nasir Uddin
Mother Name: Mst. Asma Khatun
Department : CSE
Enter Department: CSE
Student Intake: 44
Student Section: 12
ROLL : 458
SGPA: | 4.00 || 4.00 || 4.00 || 3.90 || 3.95 || 3.80 || 3.85 || 3.70 || 3.90 || 3.67 || 3.80 || 3.79 |
CGPA : 3.86
-----

```

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.

```

Student Management System || BUBT

===== Student Management System =====

Enter Roll Number: 458

NAME : MD. Shariar Rahman Oion
Father Name: MD. Nasir Uddin
Mother Name: Mst. Asma Khatun
Department : CSE
Enter Department: CSE
Student Intake: 44
Student Section: 12
ROLL : 458
SGPA: | 4.00 || 4.00 || 4.00 || 3.90 || 3.95 || 3.80 || 3.85 || 3.70 || 3.90 || 3.67 || 3.80 || 3.79 |
CGPA : 3.86

-----

Show another student information? (Y/N)?

```

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.

```

Student Management System || BUBT

===== Student Management System =====

Enter Roll number of Student to Delete the Record
Roll No. : 465

Record Deleted for

Syed Samiul Houqe
Syed Pahalobi Houqe
6421112

-----

Press any key to continue . . .






```

Fig: 6.8 Delete

Chapter 7

7.1 Coding Optimization:

Optimization refers to increase in the throughput of the system with minimal use of effort and system resources. It also includes faster response through better system design practices including optimal utilization of code, normalized database structures and their use in the code. The other criteria that ensure the optimization of this software application are enlisted as under

-  Use of modular design structure of application development.
-  Use of Normalized Database Structures-Database Optimization.
-  Use of code modules extensively-Code Optimization.
-  Use of minimum number of code lines for maximum processing of input data.
Allocating / de-allocating memory for the objects used in the code as and when required. – Garbage collection
-  Re-use of similar input forms/output reports wherever necessary to reduce the overall size of the application.

We find optimal solution of a project that is cost effective as well as time effective. Both are equally important for a project we divide the project into modules so that we can easily understand the project. And also it will take less time to solve the problems. Modularity is the single attribute of software that allows a program to be intellectually manageable.

Chapter 8

Testing

8.1 Introduction of Testing

Testing objectives: Testing is mainly done for rectifying the error from the program that is design for particular problem.

- ✚ Testing is a process of executing a program with the intent of finding an error.
- ✚ A good test case is one that has a high probability of finding an as-yet UN discovered error.
- ✚ A successful test is one that uncovers an as-yet undiscovered error.

Testing Principle: Before doing the Testing some point kept in mind

- ✚ All tests should be traceable to customer requirement.
- ✚ Tests should be planned long before testing begins.
- ✚ Testing should be begin in “small” and progress toward large.
- ✚ Exhaustive testing is not possible.

8.2 Testing and Debugging:

After programming the program has many logical errors we test our system program our system does not run successfully and does not achieve the user's requirement. If the user requirement cannot be fulfilled, we use the debugging tools in the project and debug our project in statements by statements and found error and correct the testing process focusing on logic internals of the software, ensuring that all statements have been tested.

8.3 Function Testing

System design may have so many functions. Each program has been defined into number of function. Each function has its own task. We can each function to perform an accurate result. We must debug each function.

Function is a block of code that performs a particular task, returns a particular value.

8.4 Structural Testing:

Each program has a structure, and contains the function, variable, controls, statement, decision-making loops. We can test program structure these are defined properly in our program. So the programmer set the structure of the program.

Condition Testing: Condition Testing is a test case design method that exercises the logical conditions contained in a program module.

Loop Testing: Loops are mainly used in all the module of the project, there are different type of loops in the project that I use.

Simple loops: In the simple loop in which the statement is executed inside the single loop.

Concatenated loops: Concatenated loops can be tested using the approach defined for simple loops, if each of the loops is independent of the other. However, if two loops are concatenated and the loop counter for loop 1 is used as the initial value for loop 2 then the loop are not independent. When the loops are not independent, the approach applied to nested loops is recommended.

For Loop:

A **for loop** is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

Syntax

The syntax of a **for** loop in C programming language is –

```
for ( init; condition; increment ) {  
    statement(s);  
}
```

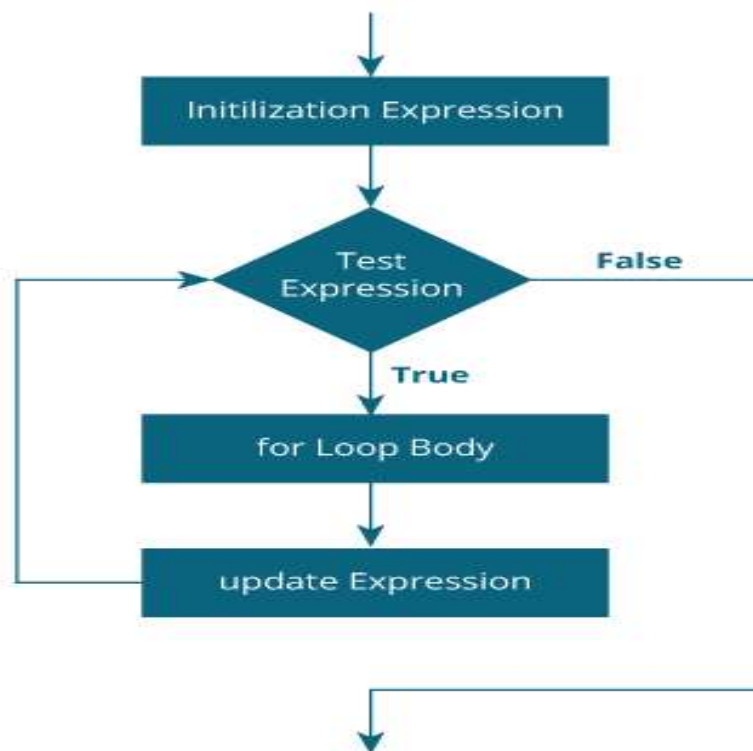


Fig: 8.1 For loop

While Loop:

A **while loop** in C programming repeatedly executes a target statement as long as a given condition is true.

Syntax

The syntax of a **while loop** in C programming language is –

```
while(condition) {  
    statement(s);  
}
```

Here, **statement(s)** may be a single statement or a block of statements. The **condition** may be any expression, and true is any nonzero value. The loop iterates while the condition is true.

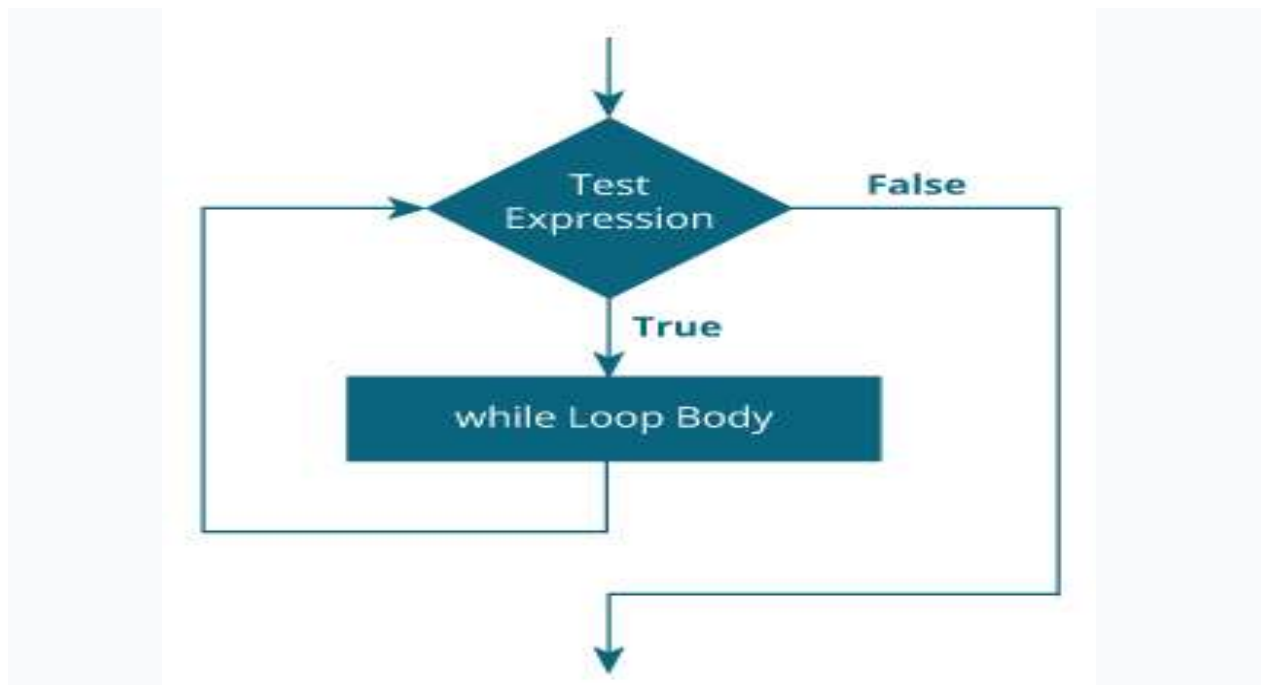


Fig: 8.2 While loop

Switch Case:

A **switch** statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each **switch case**.

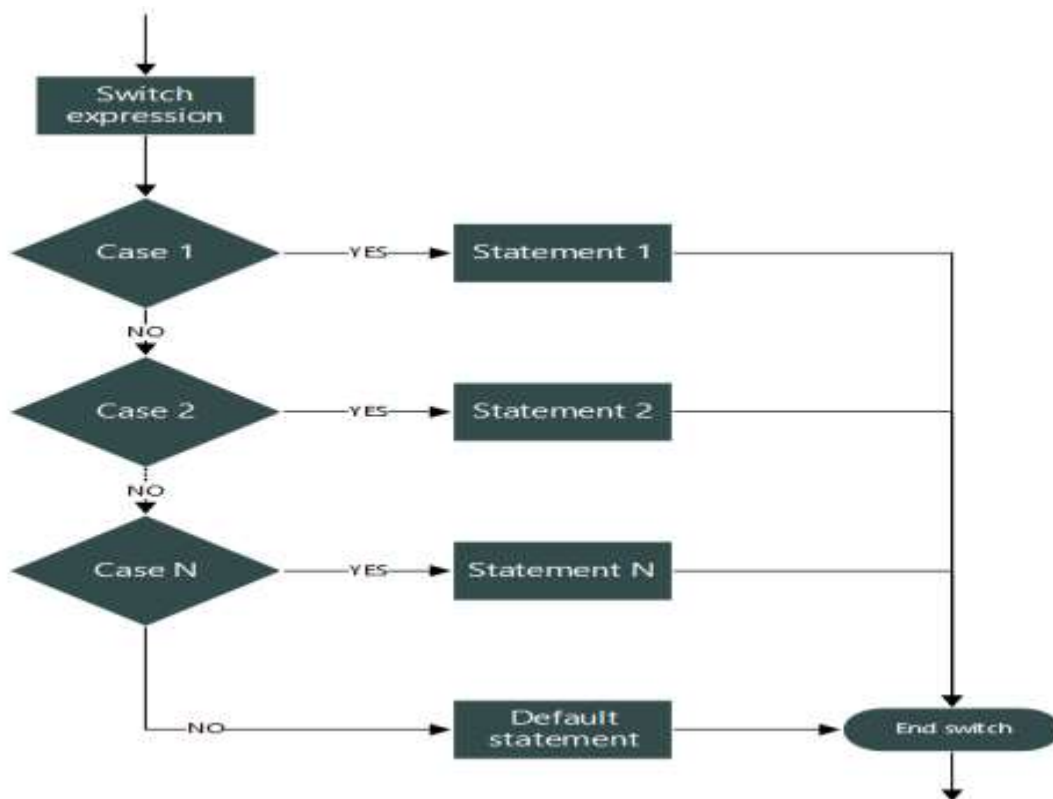


Fig: 8.3 Switch Case

Chapter 9

9.1 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 Individual(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 handling
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 handling
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[12];
    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<=1000000000; 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 Enter any key to continue.....");

```



[illegible]


```
a = getch();
if(a ==1) {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 font
Main head ()
{
    System ("Color F2");
    Loading ();
    Welcome message ();
```

```
Enter ();
List ();
}
// menu function
Invalid menu ()
{
    System ("Color F1");
    int ch, id, k, i;
    Char C, Pass [50];
    Setconsolitical ("Student Management System || BUBT");
    File * fp;
    Students s;
    int option;
    Other pirates;
    if ((fp = fopen ("db.txt", "arab +")) == null)
    {
        if ((fp = fopen ("db.txt", "wb +")) == null)
        {
            Printoff ("Cannot create or open database" );
            Return 0;
        }
    }
    When (1)
    {
        Title ();
```

```
Printoff ("n \ t");
Printer (*', 64);

Printoff ("n \ n \ t \ t \ t \ t1. Add students");
Printoff ("n \ n \ t \ t \ t \ t2. Change student");
Printoff ("n \ n \ t \ t \ t \ t3. Show all students");
Printoff ("n \ n \ t \ t \ t \ t4. Standalone view");
Printoff ("n \ n \ t \ t \ t \ t5. Remove student");
Printoff ("n \ n \ t \ t \ t \ t6. Logout n \ t");
Printer (*', 64);
printf ("n \ n \ t \ t Your t Enter your options: ->");
Scanf ("% d", & Options);

Switch (optional){
    Case 1:
        Add (FP);
        Break
    Case 2:
        Change (FP);
        Break
    Case 3:
        Display (FP);
        Break
    Case 4:
        Individual (fp);
        Break
```

Case 5:

FP = Dell (FP);

Break

Case ::

Exit (0);

Default:

printf ("ksep n \ t \ t no action detected");

Printoff ("key n \ t \ press any key n");

getch ();

System ("break");

} }

Return 1;

}

Zero printout (Church C, NTN)

{

When (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 number: ");
scanf("%d",&s.roll);
printf("\n\t\tEnter SGPA for 12 semesters\n");
for(i=0,cgpa=0; i<12; i++)
{
    scanf("%f",&s.sgpa[i]);
    cgpa+=s.sgpa[i];
}
cgpa/=12.0;
s.cgpa=cgpa;
}
cgpa/=12.0;

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

        system("pause");

        return fp;
```



```
printf("ERROR");

return NULL; }

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

printChar('-',65);

printf("\n\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\t\tRecord Found\n\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\tEnter Department: %s",s.dept);

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

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

    printf("\n\t\tStudent Roll: %d\n\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 number: ");

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

printf("\n\t\tEnter SGPA for 12 semesters\n");

for(i=0,cgpa=0; i<12; 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");

menu();}

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

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

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

        printf ("\ n \ t \ tStudent Intake:% d", s.intake);
```

```
printf ("\n \t \t Student Section:% d", s.sec);

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

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

for (i = 0; i <12; i ++ )

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

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

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

```
printf ("Department n \ t \ tEnter Department:% s", s.dept);

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

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

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

printf ("n \ tSGPA:");

for (i = 0; i <12; i ++)

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

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

printChar ('-', 65);

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

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

fflush (stdin);

another = getchar ();

}}
```


Chapter 10

10.1 REFERENCES

Website References:

<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 11

11.1 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

