PURBANCHAL UNIVERSITY

GOMENDRA MULTIPLE COLLEGE

A PROJECT REPORT ON

STUDENT RECORD MANAGER

IN PARTIAL FULFILLMENT OF THE

REQUIREMENT FOR THE DEGREE OF

BCA (1ST SEMESTER)

Submitted by

Ankush Gautam [313625]

Kabir Bhattarai [313644]

Kiran Sharma Panthi [313646]

Under the Keen Supervision of

Mr. Bhawesh Kafle

MAGH 2078

Table Of Contents

ACKNOWLEDGEMENT

According to the course of BCA first semester determined by the Purbanchal University, a computer project is to be carried out for the partial requirement for Bachelor of Computer Application (BCA). Therefore, we have developed a small 'C language' based program, "Student Record Manager".

First of all, we would like to express our sincere gratitude to Gomendra Multiple College for providing us the opportunity to pursue higher education in such a wonderful academic environment. We are indebted to Mr. Rupak Khanal (Campus Chief, G.M. College) who granted our project work by understanding the need of student and encouraging us in every aspect of our academic study.

We are grateful to our respected project teacher Mr. Bhawesh Kafle for his persistence help and clear guidance throughout our academic study and completion of this project. His suggestions and guidance in every stage is one of the major reasons for the successful completion of our project. Without his proper guidance, our project would not have been accomplished in time.

Finally, we are grateful to Purbanchal University for creating such a wonderful course framework that will undoubtedly assist students in advancing their careers. We are really glad to see a course where we can actually learn how to implement or use that knowledge in real life.

ABSTRACT

C is a popular programming language. It was developed by Dennis Ritchie in 1972 AD. Many popular software is made using this language. Therefore, this project is a small implementation of theoretical and practical knowledge of C programming. We are glad to get an opportunity to not only learn programming but also use that learning to implement and create something that can help people. So, this is a brief abstraction of what we have made and from the knowledge we got during our learning journey of C programming in BCA first semester.

This project is an eye opener for us which made us realize that no matter how well you think you know something, you will not be able to make it count unless to try to implement it and use it in practical. We learned a lot during this six-month span of our first semester but when we started to code, we came with different problems. So, we used to debug the program or search for solutions in the internet, we came to know why things are done this way. We were able to comprehend and see the concept of programming from a different perspective. In this way, we have made a project which has a special place in our heart as it is our first ever programming project.

The "Student Record Manager" undertaken as a project is based of relevant technologies. The main aim of this project is to develop software for the school and educational institute. This project has been developed to carry out the processes easily and quickly, which is not possible with the current manual system. This software is developed using C programming language in Dev C++ IDE and is meant for managing the records of employees of an educational institute. We have made the software interactive and easy to use. This software improves the performance of an institute by saving their time that used to get wasted in manual record management. This software will also help the institute to increase productivity.

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

PU	Purbanchal University
BCA	Bachelor in Computer Application
BBA	Bachelor in Business Administration
IDE	Integrated Development Environment
SRM	Student Record Manager
GM	Gomendra Multiple
MB	Megabyte
EXE	Executable Format
RAM	Random Access Memory
GPA	Grade Point Average
PC	Personal Computer

CHAPTER 1: INTRODUCTION

"Student Record Manager" is a desktop application written in C Language. The main objective of this project is to manage the details of students and calculate their marks. Besides, the program can also add, record, edit, search, delete, calculate grades and sort the details of all the students as per the requirement. We used a structure to define the attributes of a student. There are several members in the structure that defines the details of a student. For instance,

- Roll number
- Name
- Address
- Phone number
- Marks in each subject

The application begins with a setup page which asks the user to setup the program if it is the first time the Application is running. The user needs to entered the school's name, address of the school, course name (E.g., BCA, BBA), number of subjects and name of all the subjects. These details will only be asked Once. The user also needs to sign up and input the desire username and password for login after the setup and every time the program is opened. And, after the setup, the user will land on the main menu of the program. In the main menu, the user can choose from a various option according to the need of the user.

In the main menu, there are a total of 10 options the user can choose from. We used a switch case and made a function for every option on the menu for readability and to reduce redundancy. The main menu includes add record, view record, edit record, search record, delete record, sort records, view total number of records, grade-sheet, about the program and exit. The menu is inside a loop so the program won't exit unless the user hit 0 or the close the window itself.

CHAPTER 2: PROBLEM STATEMENT

Today most of the schools and educational institute use computers to calculate and store records of students. But there is few software which understand the need of college and other educational institutes. We need to manually do every thing from inputting all the details of students to writing formulas to calculate grades. From making a grade sheet design to viewing overall performance of a class.

Moreover, when it comes to calculate marks of every single student and store it, it becomes troublesome as we need different heavy software or pay subscription to use such applications. We even have to remember formulas and should know how to operate such software. Moreover, this heavy software requires a high-end devices or PC to run the software quickly and smoothly which is too expensive for a simple task. Similarly, a normal computer user can get overwhelm by the unlimited features in that software. They cannot utilize the software if they cannot properly use the features that the program contains. And a normal user needs years of practice before he/she can learn all those formulas and function to increase their productivity and speed.

Therefore, we decided to make a simple application which does this kind of job in a matter of seconds and is simple and easy to use. User doesn’t need to have a vast knowledge to use this application nor do they have to remember any kind of formulas or code to operate this program. A basic knowledge of computer is enough to operate this program and user just need to input records of students and all the other calculation and sorting things will be done automatically.

CHAPTER 3: OBJECTIVE

Due to the problems that we discussed earlier, we brainstorm what our objectives should be and we come to the conclusion that we should make something that is simple yet effective. No overwhelming features just a simple straightforward program which does what it says. Therefore, these are the objectives we wanted to achieve during our project development.

Objectives

- To provide simple and easy interface
- To provide fast operations like sorting and displaying grade-sheet
- To increase productivity and efficiency
- To provide a light-weighted free of cost program
- To provide flexibility
- To save time
- To be precise and accurate
- To be dynamic so that any school or institute can use
- Easy access of stored data
- Security

CHAPTER 4: BACKGROUND STUDY

As we already mentioned above that the use of software in school and other educational institute is increasing day by day. So, we thought of the problem regarding the bookkeeping of student's record and details. And as we had a project to make, so we decided to create a simple program for recording and calculating marks and grades of students.

We research thoroughly of what kind of feature would be useful for school and institution. And we tried to create a program as much as we know about this language which we learned for six months. So, we are bounded by the limited knowledge as it needs years of practice to master a programming language and to create a brilliant piece of art.

But we tried to do as much as we could. Our respected and dear teacher Mr. Bhawesh Kafle had already taught us basic concept and logic of adding, displaying, editing, storing information in the file. So, we kind of had a structure but we needed to have some more features and functions to make our project stand out. Therefore, we build features like sorting of records according to roll number, name or grade of students, made the program colorful and attractive, made a grade calculator, and other small feature like name checker and roll checker which would validate if the roll number or name already exists in the file.

So, we had our background study of what kind of program we are doing and made logics and functions according to it. But after we gave the project a proper structure, we got more ideas and features we could add so the process was kind of think, implement and modify and again think, implement and modify.

CHAPTER 5: REQUIREMENTS

5.1. Hardware Requirements

Minimum hardware requirements for efficient operation of this project:

• Processor	:	Pentium and above
• Hard disk	:	50 MB or above free space
• RAM	:	128 MB or above

5.2. Software Requirements

Minimum software requirements for efficient operation of this project:
As this program is written in C language, so the IDE already compiles the code and provide us an executable file. So, we don’t need any software to run this application. We don’t even need a C based IDE to run as long as we have the executable file (.exe file).

• Operating system	:	WINDOWS 98 or above
• Program Application	:	StudentRecordManager.exe

5.3. Jobwise Schedule

Member 1: Job Schedule of **Ankush Gautam**

JOB ID	JOB DESCRIPTION	DATE FROM	DATE TO
A1	Programming, Designing, Debugging	Kartik 23, 2078	Poush 30, 2078
A2	Documentation	Poush 25, 2078	Magh 10, 2078

Member 2: Job Schedule of **Kabir Bhattarai**

JOB ID	JOB DESCRIPTION	DATE FROM	DATE TO
A1	Programming	Kartik 23, 2078	Poush 30, 2078
A2	Documentation	Poush 25, 2078	Magh 10, 2078

Member 3: Job Schedule of **Kiran Sharma Panthi**

JOB ID	JOB DESCRIPTION	DATE FROM	DATE TO
A1	Programming, Debugging	Kartik 23, 2078	Poush 30, 2078
A2	Documentation	Poush 25, 2078	Magh 10, 2078

CHAPTER 6: SYSTEM DESIGN

We have designed a system where user can add record, edit record, view record, edit record etc. Therefore, we need to present a variety of option to the user. So, we have used switch case to display our main menu where user will get a various option to choose from. We designed the logic and structure of this program first with the help of Algorithm and Flowchart which gave us a clear picture of what we want to make and implement.

6.1. ALGORITHM

```
STEP 1: START
STEP 2: LOGIN ();
        a. If username and password matches, go to step 3.
        b. If username and password is invalid, go to step 2 again.
STEP 3: MENU ();
        Case 1: add record ();      break;
        Case 2: view record ();     break;
        Case 3: edit record ();     break;
        Case 4: search record ();   break;
        Case 5: delete record ();   break;
        Case 6: sort record ();     break;
        Case 7: no of records ();   break;
        Case 8: grade-sheet ();     break;
        Case 9: about ();           break;
        Case 0: exit (0);
STEP 4: Go to step 3 if Choice is not 0.
STEP 5: STOP
```

6.2. FLOWCHART

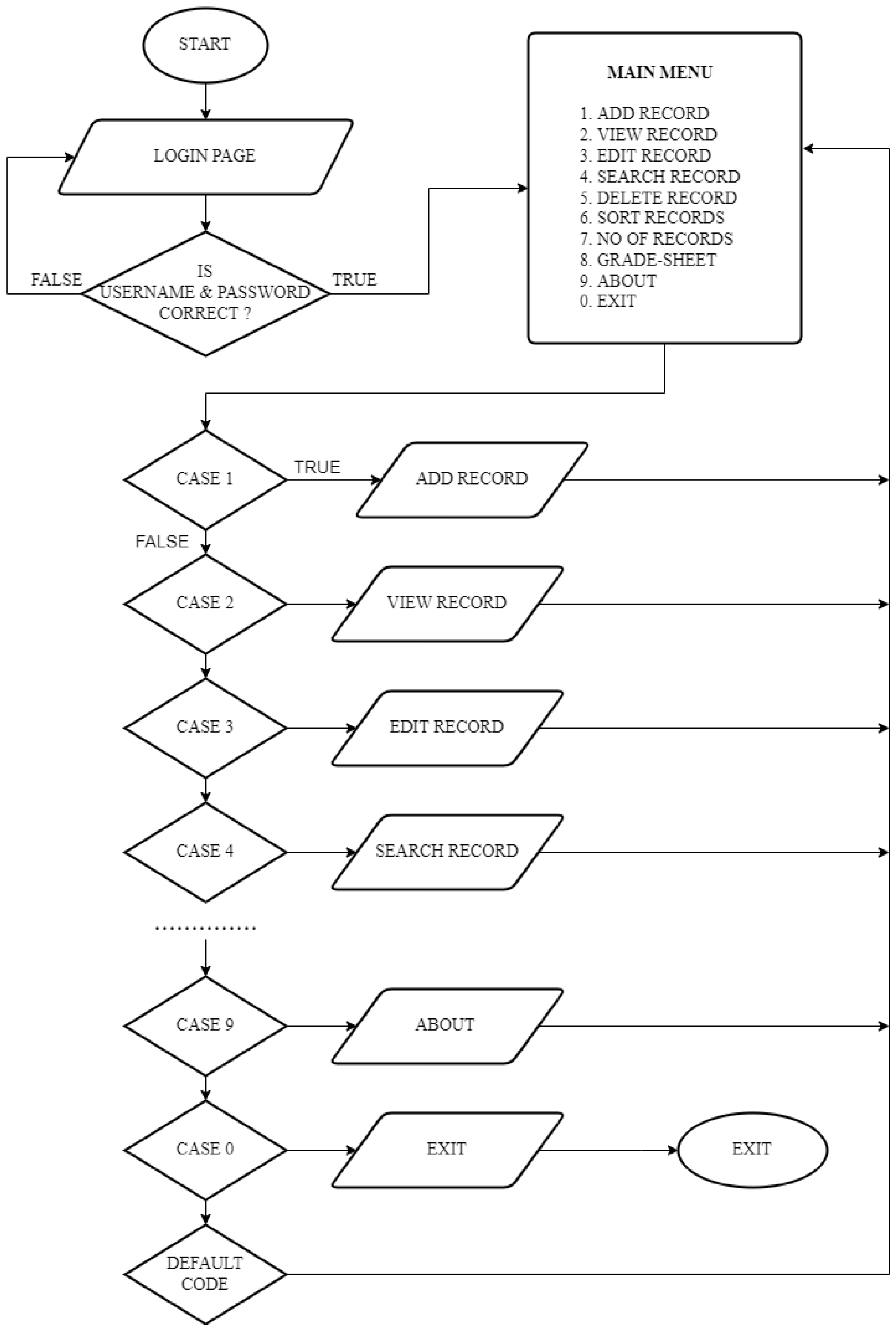


Fig 6.2. 1 : Flowchart of the Program

6.3. SOURCE CODE

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

struct setups {
    char schoolname[32], address[32], programme[32], subject[8][32];
    int semester, subs;
    struct settings {
        char username[10], password[10];
    } setting;
} setup;

struct student {
    char name[24], address[32], phone[14], grade[2];
    int roll;
    float marks[8], percent, total, gpa;
    char
} std;

void firstsetup();
void menu();
void add_record();
void view_record();
void edit_record();
void search_record();
void delete_record();
void sort_record();
int no_of_record();
void gradesheet();
void about();
void login();
char *grade_calc(float);
void inputdetails();
void menu() {
    int choice;
    do {
        header("STUDENT RECORD MANAGER"); // title of the page
        printf("\n\n\tWELCOME TO THE MAIN MENU\n\t-------------------");
        printf("\n\n\t1.ADD RECORD");
        printf("\n\t2.VIEW RECORD");
        printf("\n\t3.EDIT RECORD");
        printf("\n\t4.SEARCH RECORD");
        printf("\n\t5.DELETE RECORD");
        printf("\n\t6.SORT RECORDS");
        printf("\n\t7.NO OF RECORDS");
        printf("\n\t8.GRADE-SHEET");
        printf("\n\t9.ABOUT");
        printf("\n\t0.EXIT");
        printf("\n\n\tEnter your choice:\n\t>> ");
        scanf("%d", &choice);

        switch (choice) {
        case 1: add_record();       break;
        case 2: view_record();      break;
        case 3: edit_record();      break;
        case 4: search_record();    break;
        case 5: delete_record();    break;
        case 6: sort_record();      break;
        case 7: no_of_records();    break;
        case 8: gradesheet();       break;
        case 9: about();            break;
        case 0: exit(0);
        }
    } while (choice != 0);
}
```

```
//Add record
void add_record() {
   header("ADD RECORD");
   FILE *fadd = fopen("record.dat", "ab");
   int i,  n = no_of_record();
   char yesno;
   printf("\n\tRecord No. %d", n + 1);
   inputdetails();
   fwrite(&std, sizeof(struct student), 1, fadd);
   fclose(fadd);
   printf("\n\tRecord successfully stored!\n\t--------------------\n");
   printf("\tDo you want to add more?(Press 'y' for Yes)\n\t>> ");
   scanf(" %c", &yesno);
   if (yesno == 'y' || yesno == 'Y')
      add_record();
   else
      menu();
}
void view_record() {
   header("VIEW RECORD");
   int filesize = no_of_record();
   FILE *fview = fopen("record.dat", "rb");
   if (fview == NULL || filesize == 0) {
      printf("\n\tFile doesn't exist!Please ADD RECORD first.\n\n\t");
      system("pause");
      menu();
   }
   printf("\nROLL%-30sTOTAL\tPERCENT%-12sGRADE\n","NAME", "GPA");
   while (fread(&std, sizeof(std), 1, fview)) {
printf("\t %02d\t%-30s%06.2f\t\t%6.2f %%\t%-14.2f%s\n",  std.roll,  std.name, std.total,
std.percent, std.gpa, std.grade);
   }
   fclose(fview);
   system("pause");
}
```

```
//search-menu
void search_record() {
    header("SEARCH RECORD");
    char name[24];
    int j, found = 0 ,n = no_of_record();
    FILE *fp = fopen("record.dat", "rb");
    if (fp == NULL || n == 0) {
        printf("\n\tFile doesn't exist!Please ADD RECORD first.\n\n\t");
        system("pause");
        menu();
    }
    printf("\n\n\tEnter the NAME of the Student: ");
    scanf(" %[^\n]", name);
    while (fread(&std, sizeof(std), 1, fp)) {
        if (strcmp(name, std.name) == 0) {
            found = 1;
            printf("\n\n\tPersonal Details\n\t----------------\n");
             printf("\t%-10s %s\n\t%-10s %s\n\t%-10s %s\n", "NAME", std.name,
             "ADDRESS", std.address, "PHONE", std.phone);
            printf("\n\n\tAcademic Details\n\t----------------\n");
             printf("\t%-10s %.2f\n\t%-10s %.2f\n\t%-10s %s\n", "TOTAL", std.total, "GPA",
             std.gpa, "GRADE", std.grade);
            printf("\n\n\t%-30s MARKS\n", "SUBJECT");
            printf("\t-----------------------------------\n");
            for (j = 0; j < setup.subs; j++) {
                printf("\t%-30s%6.2f\n", setup.subject[j],std.marks[j]);
            }
        }
    }
    if (!found) {
        printf("\n\tNo Match Found!\n\t----------------");
    }
    printf("\n\t");
    system("pause");
}
```

```
void delete_record() {
    header("DELETE RECORD");
    char yesno;
    int rollno, found = 0, record_count = no_of_record();
    FILE *tempfile = fopen("tempfile.dat", "wb");
    FILE *fp = fopen("record.dat", "rb");
    if (fp == NULL || record_count == 0) {
        printf("\n\tFile doesn't exist! Please ADD RECORD first.\n\n\t");
        system("pause");
        menu();
    }
    printf("\\tROLL%-30sTOTAL\t\tPERCENT\t%-12sGRADE\n", "NAME", "GPA");
    while (fread(&std, sizeof(std), 1, fp)) {
             printf("\t%02d\t%-30s%06.2f\t\t%6.2f %%\t%-14.2f%s\n", std.roll, std.name,
           std.total, std.percent, std.gpa, std.grade);
    }
    rewind(fp);
    printf("\n\tEnter the ROLLNUMBER of the Student to DELETE: ");
    scanf("%d", &rollno);
    while (fread(&std, sizeof(std), 1, fp)) {
        if (rollno == std.roll)
            found = 1;
        else
            fwrite(&std, sizeof(std), 1, tempfile);
    }
    fclose(fp);
    fclose(tempfile);
    if (found) {
        fp = fopen("record.dat", "wb");
        tempfile = fopen("tempfile.dat", "rb");
        while (fread(&std, sizeof(std), 1, tempfile)) {
            fwrite(&std, sizeof(std), 1, fp);
        }
        fclose(fp);
        fclose(tempfile);
        printf("\n\tRecord DELETED Successfully.");
    } else {
        printf("\n\tNo Match Found!\n\t----------------\n");
    }
    printf("\tDo you want to DELETE Again?(Press 'y' for Yes)\n\t>> ");
    scanf(" %c", &yesno);
    if (yesno == 'y' || yesno == 'Y')
        delete_record();
    else
        menu();
}

//login - menu
void login() {
    header("LOGIN PAGE");
    char userinput[10], pass[10];

    printf("\n\n\tEnter Username: ");
    scanf(" %[^\n]", userinput);
    printf("\tEnter Password: ");
    scanf(" %[^\n]", pass);
    if ((strcmp(setup.setting.username, userinput) == 0) &&
(strcmp(setup.setting.password, pass) == 0)) {
        menu();
    } else {
        printf("\n\tInvalid Username or password!\n\t");
        system("pause");
        login();
    }
}
```

```
void gradesheet() {
  header("GRADE-SHEET");
  int rno, j, found = 0, n = no_of_record();
  FILE *fp = fopen("record.dat", "rb");
  printf("\n\tEnter the ROLLNUMBER of the Student: ");
  scanf("%d", &rno);
  while (fread(&std, sizeof(std), 1, fp)) {
    if (rno == std.roll) {
      found = 1;
      header("GRADE SHEET");
      printf("\n\t%s\n\t%s ", strupr(setup.schoolname) , strupr(setup.address));););
       printf("\n\n\tTHE GRADE(S) SECURED BY %s %s", strupr(std.name),
       strupr(std.address));
       printf("\n\tROLL NO: %d\tPROGRAMME: %s\t\tSEMESTER: %d\n\n", std.roll,
       setup.programme, setup.semester);
       printf("\tSN\t%-30sCREDIT\tMARKS\tGRADE\tGRADE-POINT\n",
       "SUBJECT");            for (j = 0; j < setup.subs; j++) {
       printf("\t%02d\t%-30s\t4.0\t%.2f\t  %s\t   %.2f\n", j + 1, setup.subject[j],
       std.marks[j], grade_calc(std.marks[j]), std.marks[j] / 25);
}
printf("\n\tGRADE POINT AVERAGE (GPA) : %.2f\n", std.gpa);
printf("\1.One Credit Hour equals 32 Clock Hours.\n\t2.TH : THEORY\tPR :
PRACTICAL");
printf("\n\t3.*@ : Absent\n");
printf("\n\n\tCHECKED BY:\n\t%s,%s\n", setup.schoolname, setup.address);
    }
  }
  if (!found) {
    printf("\n\tNo Match Found!\n\t----------------");
  }
  menu();
}
```

```
void firstsetup() {
   header("SETUP");
   FILE *fsetup = fopen("setup.dat", "wb");
   printf("\n\n\tPlease Take a Moment to Setup.\n");
   printf("\tEnter School Name: ");
   scanf(" %[^\n]s", setup.schoolname);
   printf("\tEnter Address: ");
   scanf(" %[^\n]s", setup.address);
   printf("\tEnter Programme (For e.g: BCA, BBA): ");
   scanf(" %[^\n]s", setup.programme);
   printf("\tEnter semester (For e.g: 1,2,3): ");
   scanf("%d", &setup.semester);
subs_again:
   printf("\tEnter No. Of Subjects: ");
   scanf("%d", &setup.subs);
   if (setup.subs > 8 || setup.subs <= 0) {
      printf("\t(Sorry! 0 < Subject <= 8)\n");
      goto subs_again;
   }
   for (int i = 0; i < setup.subs; i++) {
      printf("\tEnter Name of subject %d: ", i + 1);
      scanf(" %[^\n]", setup.subject[i]);
   }
   printf("\n\n\tSIGN UP\n\t-------\n\n");
   printf("\tEnter Username: ");
   scanf(" %[^\n]", setup.setting.username);
   printf("\tEnter Password: ");
   scanf(" %[^\n]", setup.setting.password);
   fwrite(&setup, sizeof(struct setups), 1, fsetup);
   fclose(fsetup);
   printf("\n\tSET UP SUCCESSFUL! Remember Your Details.\n\t");
   menu();
}
//END OF CODE
```

CHAPTER 7: DEVELOPMENT

7.1. Header Files

i. stdio.h
ii. conio.h
iii. string.h
iv. stdlib.h

7.2. Structures

i.
```
struct student {
        char name [24];
        char address [32];
        char phone [10];
        int roll;
        float marks [10];
        float percent;
        float total;
        float gpa;
        char grade [2];
} std;
```

ii.
```
struct setups {
        char schoolname [32];
        char address [32];
        char programme [32];
        int semester;
        int subs;
        char subject [10] [30];

        struct settings {
                char username [10];
                char password [10];
        } setting;
} setup;
```

7.3. User-Defined Functions

1. Void firstsetup ();

In this function, the user needs to provide all the necessary details for the system to work correctly. This setup function only run when the program is run for the first time ever. User needs to input details like School name, School Address, Number of subjects and Name of all the subjects. This function will create a file named "setup.dat".

2. Void Login ();

In this function, the user needs to provide username and password, he/she had entered while setting up in the setup () function. If the username or password matches, the user is directed to the menu. And, if the Username or Password is Incorrect, the user has to try again.

3. Void Menu ();

This is the Center of the program. User has a list of possible option to choose from. User cannot enter number other than 0 to 9. If entered invalid option, program will be redirected to Menu again. The program will never exit unless the user entered Number '0' or close the window.

4. Void Add Record ();

In this function, the program takes details of student and store it in the file. The details include Full name, roll number, phone, Address, marks in all the subjects. This is the first option in the menu as it is the integral part in adding and storing records.

5. Void View Record ();

In this function, the User can view the records of students that are stored in the file. User can view roll number, name, total marks and GPA. This function provides overview of the entire records that is present in the file.

6.Void Edit Record ();

In this function, the User can edit and modify the details of the student by entering the student's Roll number. The user can change any details of a student either it is personal or academic.

7.Void Search Record ();

In this function, the User can Search and View the details of a student by entering the student's Name. The Search Menu display the personal as well as academic details of a student including his/her marks in each individual subjects.

8.Void Delete Record ();

In this function, the User can delete and remove the record of any students by entering the student's Roll number. The Deleted record cannot be recovered again so user must be careful whether the record is important or not.

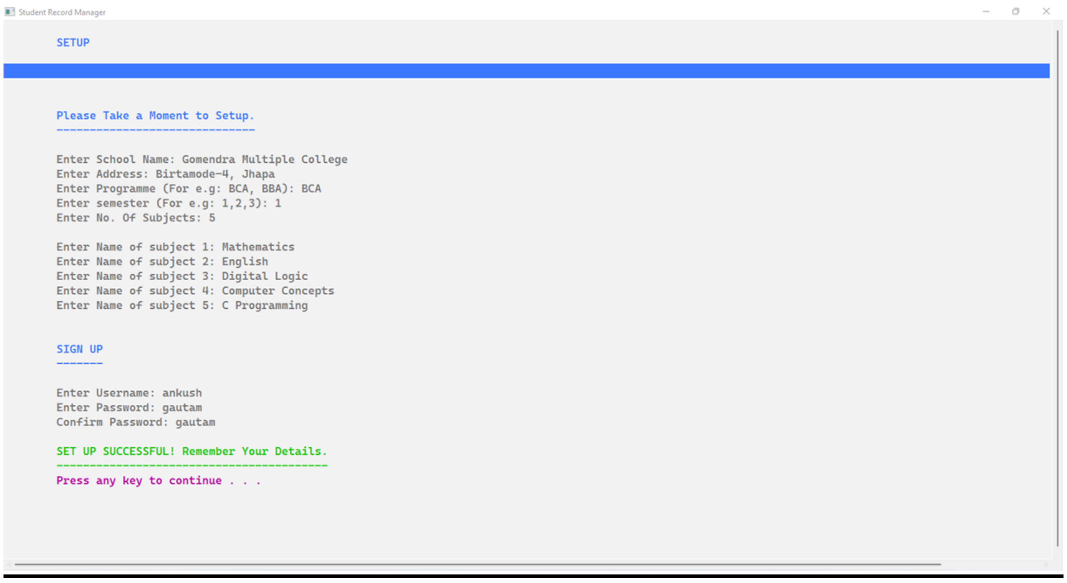
9.Void Sort Record ();

In this function, the User can sort the records of students according to Name or GPA or Roll number. User can also view the records in spread-sheet. This Function has a sub-menu for the sorting choice.

10.Void Grade-Sheet ();

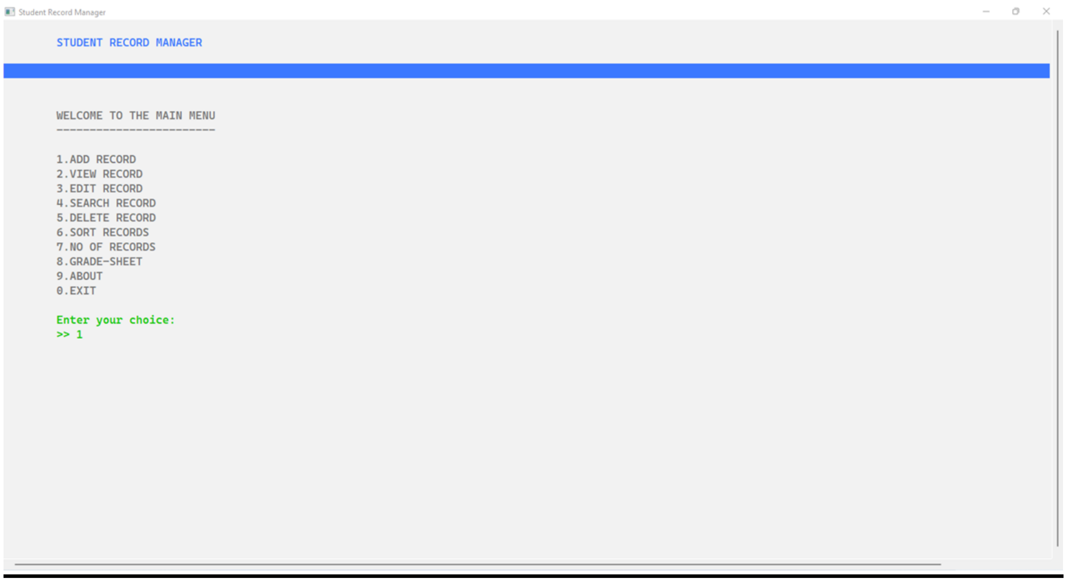
In this function, the User can view the grade-sheet of any recorded student by entering the student's Roll number. The Grade-sheet include marks of student and overall record of a student in proper format.

7.4. Screenshots Of Program



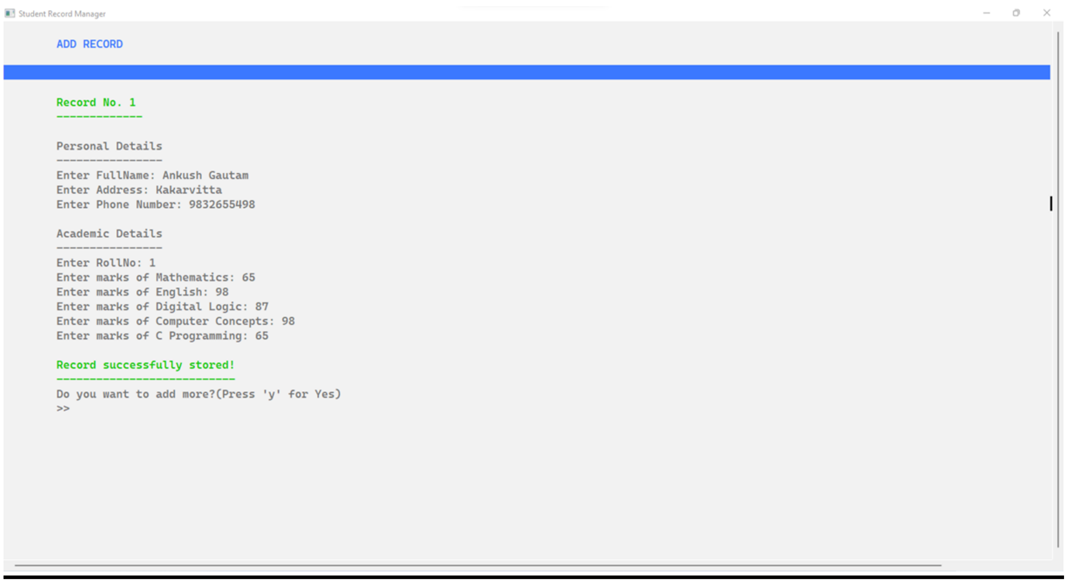
```
Student Record Manager

SETUP

Please Take a Moment to Setup.
------------------------------

Enter School Name: Gomendra Multiple College
Enter Address: Birtamode-4, Jhapa
Enter Programme (For e.g: BCA, BBA): BCA
Enter semester (For e.g: 1,2,3): 1
Enter No. Of Subjects: 5

Enter Name of subject 1: Mathematics
Enter Name of subject 2: English
Enter Name of subject 3: Digital Logic
Enter Name of subject 4: Computer Concepts
Enter Name of subject 5: C Programming


SIGN UP
-------

Enter Username: ankush
Enter Password: gautam
Confirm Password: gautam

SET UP SUCCESSFUL! Remember Your Details.
-----------------------------------------
Press any key to continue . . .
```

Fig 7.4. 1 : Setup Page



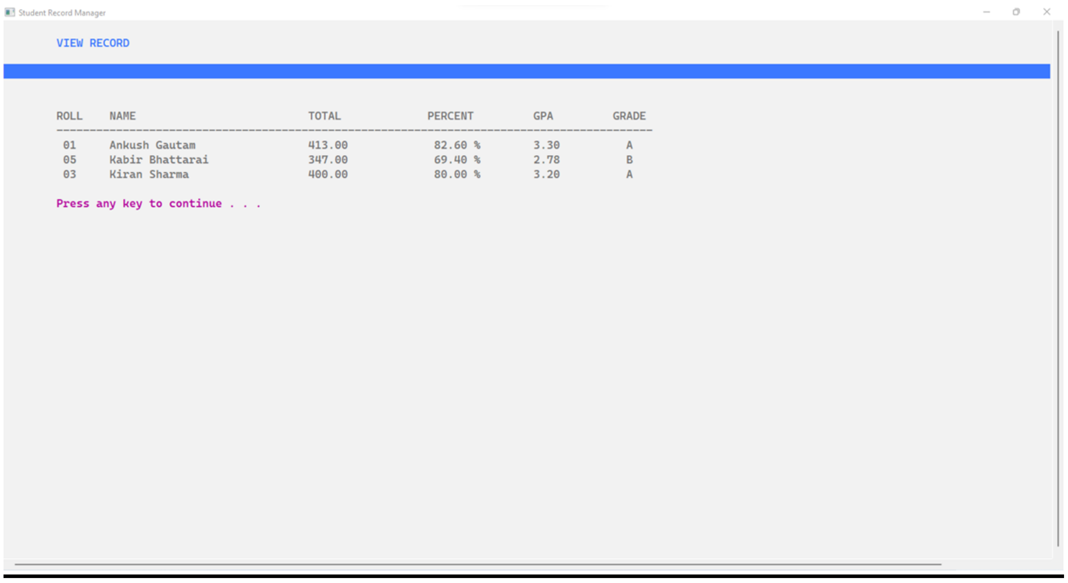
```
Student Record Manager

STUDENT RECORD MANAGER

WELCOME TO THE MAIN MENU
------------------------

1.ADD RECORD
2.VIEW RECORD
3.EDIT RECORD
4.SEARCH RECORD
5.DELETE RECORD
6.SORT RECORDS
7.NO OF RECORDS
8.GRADE-SHEET
9.ABOUT
0.EXIT

Enter your choice:
>> 1
```

Fig 7.4. 2 : Main Menu



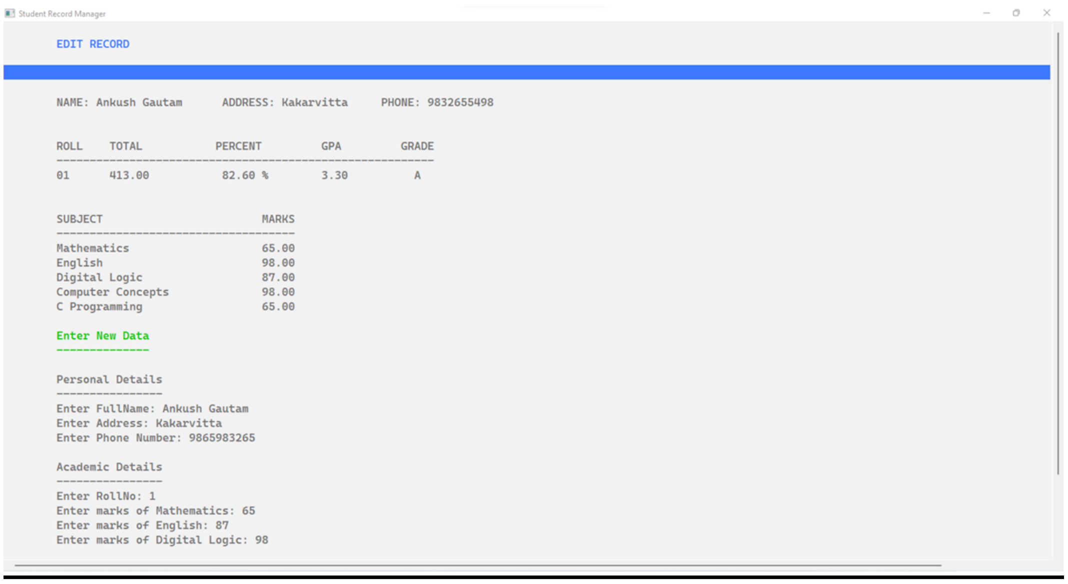
```
Student Record Manager

ADD RECORD

Record No. 1
------------

Personal Details
----------------
Enter FullName: Ankush Gautam
Enter Address: Kakarvitta
Enter Phone Number: 9832655498

Academic Details
----------------
Enter RollNo: 1
Enter marks of Mathematics: 65
Enter marks of English: 98
Enter marks of Digital Logic: 87
Enter marks of Computer Concepts: 98
Enter marks of C Programming: 65

Record successfully stored!
---------------------------
Do you want to add more?(Press 'y' for Yes)
>>
```

Fig 7.4. 3 : Add Record



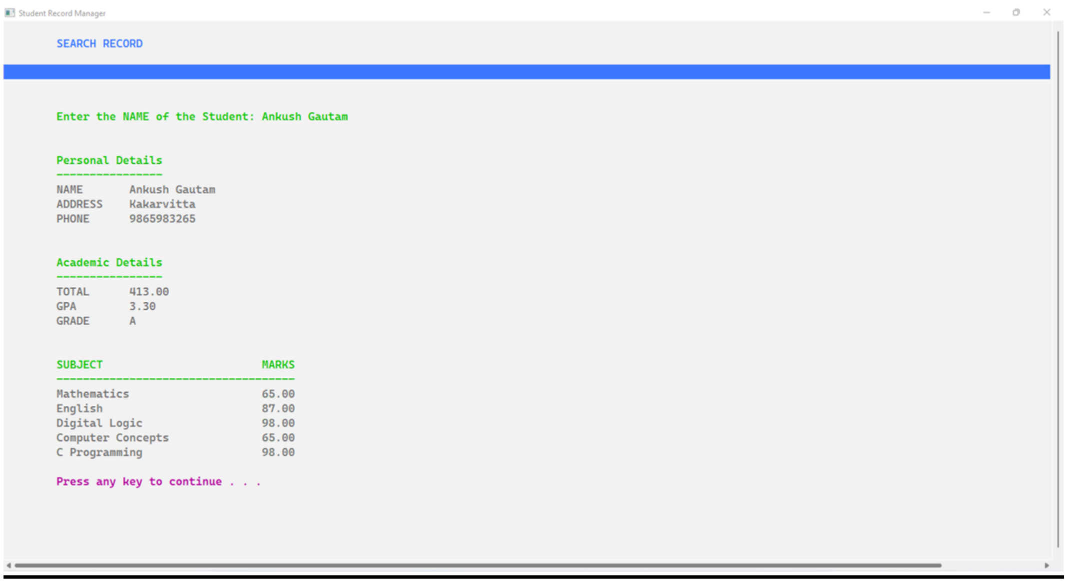
```
Student Record Manager

VIEW RECORD

ROLL    NAME                          TOTAL             PERCENT         GPA         GRADE
------------------------------------------------------------------------------------------
 01     Ankush Gautam                 413.00             82.60 %        3.30          A
 05     Kabir Bhattarai               347.00             69.40 %        2.78          B
 03     Kiran Sharma                  400.00             80.00 %        3.20          A

Press any key to continue . . .
```

Fig 7.4. 4 : View Record



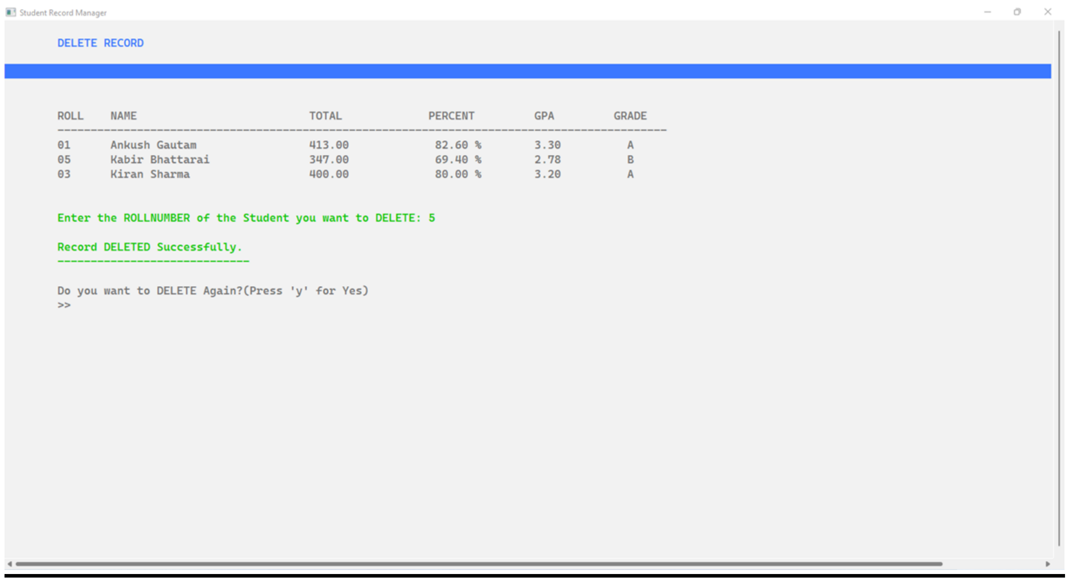
```
Student Record Manager

EDIT RECORD

NAME: Ankush Gautam     ADDRESS: Kakarvitta     PHONE: 9832655498

ROLL    TOTAL           PERCENT         GPA         GRADE
---------------------------------------------------------
01      413.00           82.60 %        3.30          A

SUBJECT                         MARKS
-------------------------------------
Mathematics                     65.00
English                         98.00
Digital Logic                   87.00
Computer Concepts               98.00
C Programming                   65.00

Enter New Data
--------------

Personal Details
----------------
Enter FullName: Ankush Gautam
Enter Address: Kakarvitta
Enter Phone Number: 9865983265

Academic Details
----------------
Enter RollNo: 1
Enter marks of Mathematics: 65
Enter marks of English: 87
Enter marks of Digital Logic: 98
```

Fig 7.4. 5 : Edit Record



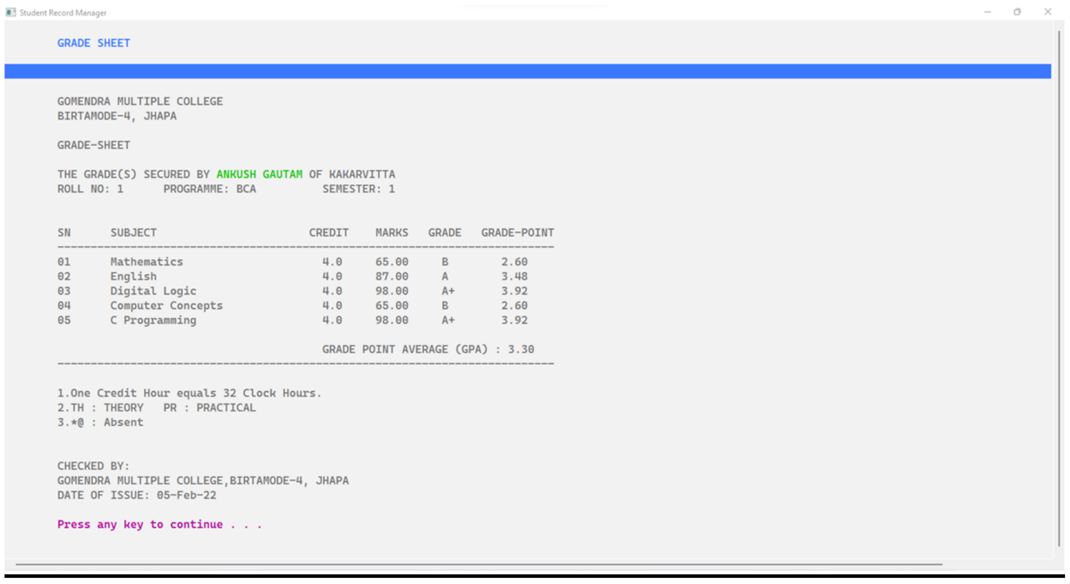
```
Student Record Manager

SEARCH RECORD

Enter the NAME of the Student: Ankush Gautam

Personal Details
----------------
NAME       Ankush Gautam
ADDRESS    Kakarvitta
PHONE      9865983265

Academic Details
----------------
TOTAL      413.00
GPA        3.30
GRADE      A

SUBJECT                         MARKS
-------------------------------------
Mathematics                     65.00
English                         87.00
Digital Logic                   98.00
Computer Concepts               65.00
C Programming                   98.00

Press any key to continue . . .
```

Fig 7.4. 6 : Search Record



```
DELETE RECORD

ROLL    NAME                          TOTAL            PERCENT        GPA         GRADE
-------------------------------------------------------------------------------------------
01      Ankush Gautam                 413.00            82.60 %       3.30         A
05      Kabir Bhattarai               347.00            69.40 %       2.78         B
03      Kiran Sharma                  400.00            80.00 %       3.20         A

Enter the ROLLNUMBER of the Student you want to DELETE: 5

Record DELETED Successfully.
-----------------------------

Do you want to DELETE Again?(Press 'y' for Yes)
>>
```

Fig 7.4. 7 : Delete Record



```
GRADE SHEET

GOMENDRA MULTIPLE COLLEGE
BIRTAMODE-4, JHAPA

GRADE-SHEET

THE GRADE(S) SECURED BY ANKUSH GAUTAM OF KAKARVITTA
ROLL NO: 1      PROGRAMME: BCA         SEMESTER: 1

SN      SUBJECT                       CREDIT    MARKS    GRADE   GRADE-POINT
---------------------------------------------------------------------------
01      Mathematics                     4.0     65.00     B         2.60
02      English                         4.0     87.00     A         3.48
03      Digital Logic                   4.0     98.00     A+        3.92
04      Computer Concepts               4.0     65.00     B         2.60
05      C Programming                   4.0     98.00     A+        3.92

                                        GRADE POINT AVERAGE (GPA) : 3.30
---------------------------------------------------------------------------

1.One Credit Hour equals 32 Clock Hours.
2.TH : THEORY   PR : PRACTICAL
3.*@ : Absent

CHECKED BY:
GOMENDRA MULTIPLE COLLEGE,BIRTAMODE-4, JHAPA
DATE OF ISSUE: 05-Feb-22

Press any key to continue . . .
```

Fig 7.4. 8 : Grade-Sheet

7.5. Files Used

1. **about.dat**
   This file includes details about the developers and the program. This file already exists in the program as it is made by us directly.

2. **setup.dat**
   This file includes the settings that the user entered while setting up the program for the first time. This includes number of subjects, name of subjects, school name, school address, programme & semester, sign up (username & password)

3. **record.dat**
   This file is created when the user records students in the program. This file includes only the details of the students. This is the most crucial file of the program

4. **tempfile.dat**
   This file is created while deleting or editing the records of a student. This a temporary file and is safe to be deleted.

CHAPTER 8: TESTING

Test Case 8. 1:

Test Case Description: LOGIN MENU

SCENARIO NO. 1	Entering Incorrect Username and Password.
ACTION	Enter Username: admin Enter Password: something
REQUIRED OUTPUT	Invalid username or password!
ACTUAL OUTPUT	Invalid username or password!
REMARKS	PASS

SCENARIO NO. 2	Entering Correct Username and Password.
ACTION	Enter Username: admin Enter Password: password
REQUIRED OUTPUT	Welcome to the Main Menu.
ACTUAL OUTPUT	Welcome to the Main Menu.
REMARKS	PASS

Test Case 8. 2:

Test Case Description: GRADE-CALCULATOR

SCENARIO	To check if grade calculator works correctly
ACTION	Total marks = 439
REQUIRED OUTPUT	GPA = 3.51 and Grade = A
ACTUAL OUTPUT	GPA = 3.51 and Grade = A
REMARKS	PASS

Test Case 8. 3:

Test Case Description: FILE HANDLING

SCENARIO	To check if details are stored in the file correctly
ACTION	Added records of a student
REQUIRED OUTPUT	Records stored in the file properly.
ACTUAL OUTPUT	Records stored in the file properly.
REMARKS	PASS

Test Case 8. 4:

Test Case Description: MAIN MENU

SCENARIO	To check if the Menu displays all the options
ACTION	Viewing main menu after logging in
REQUIRED OUTPUT	All 10 options are works fine.
ACTUAL OUTPUT	All 10 options are working fine.
REMARKS	PASS

Test Case 8. 5:

Test Case Description: SORTING RECORDS

SCENARIO	To check records are sorted correctly
ACTION	Checking sort by name, sort by GPA
REQUIRED OUTPUT	Records are sorted in a proper format
ACTUAL OUTPUT	Records are sorted in a proper format
REMARKS	PASS

CHAPTER 9: PROJECT RESULT

Along with all the above program testing we showed in the earlier chapter, we persistently tested every function of our project so that it doesn't behave unexpectedly. Many unexpected behaviors arise but we tackle it. For instance, there was a problem with file handling where every student would have same roll number but we found the solution after many hours of debugging. Every program has a bug which arises once a while so we got to be up to date with it to make our program less vulnerable. Some of results of our testing are shown below:

S.N.	TEST OBJECTIVE	RESULT
1.	To check whether the program runs or not	*Successful*
2.	To check if the menu displays all the options or not	*Successful*
3	To check if all the menu options are correct or not	*Successful*
4	To check if the record is added in the file or not	*Successful*
5	To check if the program calculates correct GPA	*Successful*
6	To check if the 'View Record' displays details or not	*Successful*
7	To check if the program returns back to Menu or not	*Successful*
8	To check if the exit option works or not	*Successful*
9	To check if the login/authentication works or not	*Successful*
10.	To check if the sorting of records is correct or not	*Successful*
11.	To check if colors are displayed correctly or not	*Successful*

Table 9. 1 : Result of Testing

CHAPTER 10: FUTURE ENHANCEMENT

No Program is free of bugs and No Program is 100% completed. There is always a room for improvement and enhancement. Similarly, this project also has some bugs and glitches that we can fix in the future. Some of the problems may be poor memory management like we could have optimize the code more so it can use every last bit of memory.

There is definitely some room for improvement in memory management. We can also make dynamic allocation of memory so we can optimize the program by allocating memory required for that time. For example: We have initialized a subject array with 8 subjects but user can choose only 5 subjects, So the remaining 3 spaces are just wasted. We could optimize and work on 2-dimensional dynamic allocation.

Moreover, we could also add different new features and functions like we can implement sorting of records by highest marks of students in a particular subject or we can implement features like multiple users can access the program at once. We can also implement multiple classes features like this program can only store records of one class or grade but we could enhance the program by adding feature that can store information of student of different or more than one class. There will always be some ideas we can implement.

But we are bound by our limited knowledge and time. So, as we grow and learn more about programming and how a program is made, we can implement or make more projects with more features. Any ideas or feedback is much appreciated.

CHAPTER 11: CONCLUSION

As this is our first ever programming project, we are very happy to see how far we have come. We are glad that we came close enough to what we have imagined before starting this project. With a learning of six months, we have implemented a decent program, so we are optimistic that with persistent learning in the world of programming we can build something that can be useful to people.

“Student Record Manager” has evolved that way we wanted and we are grateful for all the helps we got. We tried to make a simple yet effective program and we think we kind of were successful. This project is a small attempt to help educational institute to increase their productivity and provide them a free of cost program which calculates and display the result of an entire class in a matter of seconds.

As we already discussed in above chapter about further enhancement and every program has a room for further improvement so any kind of feedback is most welcomed. We are not only open to feedbacks but any constructive criticism is also welcomed with a happy face.

We would again like to thank our respected teacher Mr. Bhawesh Kafle for consistent support and help. We are also thankful to our classmate for being so creative and helpful that inspire us to push ourselves further and learn more every day.

Thank you everyone for your time.

REFERENCES

Online

- https://www.geeksforgeeks.org/c-programming-language
- https://www.tutorialspoint.com/cprogramming
- https://www.youtube.com/watch?v=ytGATjX3nqc&ab_channel=EngineerMn
- https://www.youtube.com/watch?v=L00f3z7WBRU&ab_channel=Brightgoal
- https://youtu.be/1lNL31q4T5I

Books

1. Kernighan, Brian; Ritchie, Dennis M. (March 1988). *The C Programming Language* (2nd ed.). Englewood Cliffs, NJ: Prentice Hall. ISBN 0-13-1103628

2. King, K.N. (2008). *C Programming: A Modern Approach* (2 ed.). W. W. Norton. ISBN 978-0393979503.