**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“JnanaSangama”, Machhe, Belagavi, Karnataka-590018**



**Intra / Inter-Institutional Internship Report**

***Submitted in partial requirements of Intra / Inter Internship of Third Semester***

**Bachelor of Engineering**

in

**Computer Science and Engineering**

Submitted by

**ANWITHA 4GW21CS016**

**DHANYASHREE TN 4GW21CS031**

**HARSHITHA KR 4GW21CS035**

Under the Guidance of

**RAJATH A N**

Assistant Professor

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**(Accredited by NBA, New Delhi, Validity: 01.07.2017 – 30.06.2020 & 01.07.2020 – 30.06.2023)**

**GSSS INSTITUTE OF ENGINEERING & TECHNOLOGY FOR WOMEN**

(Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi & Govt. of Karnataka)

(Accredited with Grade ‘A’ by NAAC)

K.R.S Road, Metagalli, Mysuru-570016, Karnataka

**2022-2023**

***GeethaShishuShikshanaSangha (R)***

**GSSS INSTITUTE OF ENGINEERING & TECHNOLOGY FOR WOMEN**

(Affiliated to VTU, Belagavi, Approved by AICTE -New Delhi & Govt. of Karnataka)

(Accredited with Grade ‘A’ by NAAC)

K.R.S Road, Mysuru-570016, Karnataka

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**(Accredited by NBA, New Delhi, Validity: 01.07.2017 – 30.06.2020 & 01.07.2020 – 30.06.2023)**

****

**CERTIFICATE**

Certified that the Intra / Inter-Institutional Internship is a bonafide work carried out by **Anwitha(4GW21CS016),DhanyashreeT.N(4GW21CS031), andHarshitha K.R (4GW21CS035)** in partial fulfilments or the award of the degree of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi during the year 2022-2023. The Inter/Intra Institutional Internship project has been approved as it satisfies the academic requirements with respect to the Intra / Inter-Institutional Internship project prescribed for a Bachelor of Engineering Degree.

*Signature of the Guide Signature of the HOD*

**Rajath A N Dr. S MeenakshiSundaram**

Assistant Professor Professor& Head

**ACKNOWLEDGEMENT**

We sincerely owe my gratitude to all the persons who helped and guided me in completing this Intra / Inter-Institutional Internship.

We are thankful to **Mrs. Vanaja B Pandit**, *Honorary Secretary*, GSSSIETW, Mysuru, for having supported in my academic endeavours.

We are thankful to **Dr.Shivakumar M**, *Principal*, GSSSIETW, Mysuru, for all the support he has rendered.

We thank **Dr. S MeenakshiSundaram**, *Professor and Head*, Department of Computer Science and Engineering, for his constant support and encouragement throughout the tenure of this Inter/Intra Institutional Internship work.

We would like to sincerely thank my guide **Rajath A N,***Assistant Professor*, Department of Computer Science and Engineering, for providing relevant information, valuable guidance and encouragement to complete this Intra / Inter Institutional Internship.

We are extremely pleased to thank my parents, family members and friends for their continuous support, inspiration and encouragement, for their helping hand and also last but not the least, we thank all the members who supported directly or indirectly in the academic process.

**ANWITHA**

**DHANYASHREE TN**

**HARSHITHA KR**

nba-logo1.png

**KRS ROAD, METAGALLI | MYSURU – 570 016 | KARNATAKA | INDIA**

***(Affiliated to Visvesvaraya Technological University, Belagavi, Approved by AICTE New Delhi & Govt of Karnataka)***

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

|  |  |  |
| --- | --- | --- |
| **Date(s)** | **Intra / Inter Internship Details** | **Duration**  **in hours** |
| **11.10.2022**  **Tuesday** | Speed Math 1 – Multiplication, Speed Math 2 – Multiplication, Stress Management | **7** |
| **12.10.2022**  **Wednesday** | Speed Math 3 - Addition and Subtraction, Speed Math 4 - Division and Fractions, Leadership Skills | **7** |
| **13.10.2022**  **Thursday** | Mind Mapping and SWOT Analysis, Speed Math 5 - Decimals & Percentages, Goal Setting, Individual Presentations, Individual Presentations, Speed Math 4 - Division and Fractions | **7** |
| **14.10.2022** | Individual Presentations, Individual Presentations, Number System, Speed Math 5- Decimals & Percentages, Number System, Goal Setting | **7** |
| **15.10.2022**  **Saturday** | HCF & LCM, Decimal Fractions, Simplification, Essay Writing, HCF & LCM, Decimal Fractions, Essay Writing, Simplification | **7** |
| **17.10.2022**  **Monday** | Introduction to internship, Programming Proficiency Test to know your Programming Skill Level in ‘C’, Discussion of answers related to Programming Proficiency Test Formation of Project Groups & Batch wise Allotment of Projects in ‘C’ | **7** |
| **18.10.2022**  **Tuesday** | Tips and Tricks of ‘C’ Programming & Project Management Skills, Project Development – Algorithm Design and Coding | **7** |
| **19.10.2022**  **Wednesday** | Overview of Project Management Plan & Project Coding, Review of Project Codes & Guidance for Efficient Coding | **7** |
| **20.10.2022**  **Thursday** | Overview of Project Management Plan & Project Coding, Review of Project Codes & Guidance for Efficient Coding | **7** |
| **21.10.2022**  **Friday** | Project Testing with Sample Inputs and Output Verification   * Project Report Preparation and Discussions * Delivery of sample project format to project batches | **7** |
| **Total Hours** | | **70** |
| **22.10.2022 to**  **30.10.2022** | Intra / Inter Internship Project Report Preparation | |
| **31.10.2022**  **Monday** | Submission of Internship Report by the students to the Department | |

**Intra / Inter Internship Schedule**

(Carried our during the intervening vacation period of II and III Semester)

**TABLE OF CONTENTS**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  | **Acknowledgment**  **Intra / Inter Internship Schedule** | **3**  **4** | | **1** | **Abstract**  **1.1 Objectives of The Project**  **1.2 Outcomes of The Project** | **6**  **7**  **7** | | **2** | **Methodology** | **8** | | **3** | **Flow Chart** | **11** | | **4** | **Algorithm** | **12** | | **5** | **Source Code** | **13** | | **6** | **Sample Inputs & Outputs** | **37** | | **7** | **Conclusions And Future Scope** | **40** | | **8** | **REFERENCES** | **41** | |  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NUMBER** | **DESCRIPTION** | **PAGE NUMBER** |
| 1 | Dataflow diagram | 10 |
| 2 | Flowchart | 11 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**STUDENT RESULT MANAGEMENT SYSTEM**

**ABSTACT**

We all know the importance of education in one’s life which has led to revolution in field of education. Universities, colleges, schools have loads of task to be completed in given timeline. In today’s scenario colleges needs to student result manually which takes loads of time and effort by faculties working on it. Hence in order to simplify this task a web base system is introduced which can perform student result analysis.

The STUDENT RESULT MANAGEMENT SYSTEM allows the student to check, view, and see their results. The main objective of the student result management system is to manage the grades/results of various examinations, courses, etc. It manages all the details about the result, exam, student, activity, grade, etc. It is a process of managing and controlling. It is also a technological opportunity for schools, colleges and universities for simple and secure results.

The code is developed for a system that will manage information about the various users, subjects offered in various semesters, marks obtained by students in semesters, and the generation of reports.

The project aims to provide student results in an effortless way. Here, we make use of Data structure which is a named location that can be used to store and organize data. A linked List Data Structure is implemented which consists of nodes where each node contains a data field and a reference (link) to the next node in the list.

This code helps to perform various operation on the result. It stores the result and information of the student in database safely, when user wants to view then it will easily retrieve the information from the database.

**OBJECTIVES OF THE PROJECT**

1. Manage the information of the students.

2. Managing and displaying the results.

3. Tracks the details about subjects and students.

4. Semester-wise results can be shown.

5.Analysing the results.

**OUTCOMES OF THE PROJECT**

1. Storing the student’s details like name, subject, USN etc..

2. Inserting and displaying the marks.

3. Deleting and changing the Information is possible.

4. Semester-wise results can be given.

**METHODOLOGY**

In this project, we used devC++ software to execute this program.

The project “Student Result Management System” is a student database and result management system. It takes the input from the student database file, and the subject scheme from the subject file, and writes the results into another file.

This program is used to add the student records, display student records, add new scheme, display all schemes, reads schemes from a file, delete a scheme for any changes, reads student data from the fill.

It stores input, reads and displays the marks awarded to the student so that they can analyse the marks. After this it will print the results semester wise so that students can compare the results of previous and current semester. Such that they can improve the performance in their upcoming semester exam.

Different concepts are used in the code such as Structures, Pointers, Switch statements, Looping statements, and conditional statements. Storage of data is done using Linked List concept.

We have declared the header file according to our programrequirements. Struct and typedef are used to declare the student records like name, roll number, semester, subject, etc…Pointer is used to store the address of the data in the “student.txt” file.

With the help of the switch statement, we have included the twelve cases to read, store and display the student data. These cases are shown in the flowchart. When we run this program, it will give the option to choose the case. According to the chosen case, the function is executed and displays the result.

The Case Description

After the execution of the switch statement, we have to select a case. If the

Case 1: Insertion of the student data takes place.

Case 2: Display of student data.

Case 3: Add the new scheme.

Case 4: Display all schemes.

Case 5: Read schemes from the file.

Case 6: Delete a scheme if needed.

Case 7: Read student data.

Case 8: Input marks awarded.

Case 9: Read marks awarded.

Case 10: Display marks awarded.

Case 11: Print results semester-wise.

Case 12: Random access of a record.

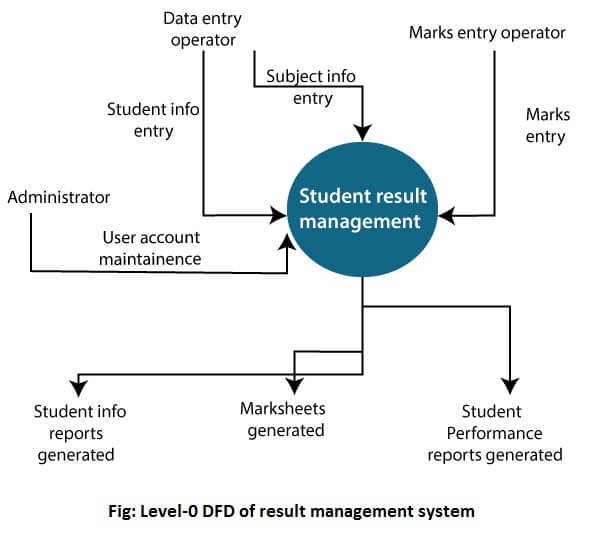
These cases are executed according to the case number we give as input.

This project is all about database management system of students result. The student database management is **designed to plan, organize, and maintain students' learning outcomes**. This application, accessible in every LEAD-powered school, helps organize documentation, admission details, administration, and other relatable processes.

How does this help the colleges

1. Manage examination results easily.
2. Manage student data.
3. Generate automatic results smoothly.

DATA FLOW DIAGRAM



**FLOW CHART**

Add record

Switch(ch)

Case 1

Input

Display record

Case 2

Add new scheme

Case 3

Display scheme

Case 4

Case 5

Read scheme

Delete a scheme

Case 6

Read record

Case 7

Case 8

I/P marks

Read marks

Case 9

Case10

Display marks

Case11

Print marks

Access record

Case12

output

**ALGORITHM**

Step 1: Start

Step 2: Read the structure data like name, roll number, subject,etc..

Step 3:While(1)

{

Switch(ch)

{

Case 1:read student data function is executed

Break;

Case 2: display student data function is executed

Break;

Case 3: Add new scheme function is executed

Break;

Case 4: Display all scheme function is executed

Break;

Case 5: Read scheme from the file function is executed

Break;

Case 6: Delete a scheme function is executed.

Break;

Case 7:Read student records from file function is executed.

Break;

Case 8:Input marks awarded function is executed.

Break;

Case 9:Read marks awarded function is executed.

Break;

Case 10: Display marks awarded function is executed.

Break;

Case 11: Print result semester wise function executed.

Break;

Case 12: Random access of record.

Break;

}

}

Step 4: Function which are outside the main function is according to

Switch statement cases.

Step 5: Display the student record.

Step 6: Stop.

**SOURCE CODE**

#include <iostream>

#include<iomanip>

#include<malloc.h>

#include<process.h>

#include<fstream>

#include<string.h>

#include<stdio.h>

using namespace std;

structscheme\_l //linked list for scheme

{

char sub\_code[10];

char sub\_title[20];

intsess\_marks;

inttheo\_marks;

intsem;

structscheme\_l\* next;

};

typedefstructscheme\_lScnode;

Scnode\* start = NULL;

Scnode\* writeScheme(Scnode\* );

Scnode\* displayScheme(Scnode\* );

Scnode\* readScheme(Scnode\* );

Scnode\* addnode(Scnode\*,Scnode\* );

Scnode\* deleteScheme(Scnode\* );

struct student

{

introll\_no; //primary key

char name[20];

char f\_name[20];

char dob[10];

intsem;

intyoa; //year of admission

struct student\* next;

};

typedefstruct student Stnode;

Stnode\* starts = NULL;

Stnode\* writeStudentRecord(Stnode\* );

Stnode\* displayStudentRecord(Stnode\* );

Stnode\* readStudentRecord(Stnode\* );

Stnode\* addnodeStudent(Stnode\*,Stnode\* );

structrecord\_l

{

intr\_roll\_no;

char r\_name[20];

intr\_sem;

intma\_sess; //marks awarded

intma\_theo;

structrecord\_l\* next;

};

typedefstructrecord\_lRnode;

Rnode\* startl = NULL;

Rnode\* input\_marks(Rnode\* );

Rnode\* writeRecord(Rnode\*, Rnode\* );

Rnode\* readRecord(Rnode\* );

Rnode\* addnodeRecord(Rnode\*, Rnode\* );

Rnode\* displayMarksRecord(Rnode\* );

Rnode\* printRecordSemester(Rnode\* );

Rnode\* listRecord(Rnode\* );

int main()

{

int choice;

while(1)

{

cout<<"\n1.Add student records\n";

cout<<"2.Display student records\n";

cout<<"3.Add new scheme\n";

cout<<"4.Display all schemes\n";

cout<<"\t\t\t5.Read schemes from file\n";

cout<<"6.Delete a scheme\n";

cout<<"\t\t\t7.Read student records from file\n";

cout<<"8.Input marks awarded\n";

cout<<"\t\t\t9.Read marks awarded\n";

cout<<"10.Display marks awarded\n";

cout<<"11.Print result semester wise\n";

cout<<"12.Random access of a record\n";

cin>>choice;

switch(choice)

{

case 1:

starts = writeStudentRecord(starts);

break;

case 2:

starts = displayStudentRecord(starts);

break;

case 3:

start = writeScheme(start);

break;

case 4:

start = displayScheme(start);

break;

case 5:

start = readScheme(start);

break;

case 6:

start = deleteScheme(start);

break;

case 7:

starts = readStudentRecord(starts);

break;

case 8:

startl = input\_marks(startl);

break;

case 9:

startl = readRecord(startl);

break;

case 10:

startl = displayMarksRecord(startl);

break;

case 11:

startl = printRecordSemester(startl);

break;

case 12:

startl = listRecord(startl);

break;

default:

return 0;

break;

}

}

return 0;

}

Scnode\* writeScheme(Scnode\* start)

{

FILE\* fp;

fp = fopen("Scheme.txt","ab+");

Scnode\* newnode;

newnode = (Scnode\*)malloc(sizeof(Scnode));

Scnode\* ptr;

char dsub\_code[10];

char dsub\_title[20];

intdsess\_marks;

intdtheo\_marks;

intdsem;

cout<<"Enter subject code\n";

cin>>dsub\_code;

cout<<"Enter subject name\n";

fflush(stdin);

gets(dsub\_title);

fflush(stdin);

cout<<"Enter maximum marks of sessional\n";

cin>>dsess\_marks;

cout<<"Enter maximum marks of end semester theory\n";

cin>>dtheo\_marks;

cout<<"Enter semester number (1-8)\n";

cin>>dsem;

strcpy(newnode->sub\_code,dsub\_code);

strcpy(newnode->sub\_title,dsub\_title);

newnode->sess\_marks = dsess\_marks;

newnode->theo\_marks = dtheo\_marks;

newnode->sem = dsem;

if(start == NULL)

{

cout<<"Making the first node\n";

start = newnode;

newnode->next = NULL;

cout<<"Writing start to file\n";

fwrite(start,sizeof(Scnode),1,fp);

}

else

{

cout<<"Making new node\n";

ptr = start;

while(ptr->next != NULL) ptr = ptr->next;

ptr->next = newnode;

newnode->next = NULL;

cout<<"Writing node to file\n";

fwrite(ptr->next,sizeof(Scnode),1,fp);

}

fclose(fp);

return start;

}

Scnode\* readScheme(Scnode\* start) //function to read linked lists from file

{

FILE\* fq;

fq = fopen("Scheme.txt","rb+");

Scnodesc;

fread(&sc,sizeof(Scnode),1,fq);

start = displayScheme(start);

while(!feof(fq))

{

cout<<"\nYES\n";

start = addnode(start,&sc);

cout<<endl<<sc.sub\_code<<endl<<sc.sub\_title<<endl<<sc.sess\_marks<<endl<<sc.theo\_marks<<endl<<sc.sem;

fread(&sc,sizeof(Scnode),1,fq);

start = displayScheme(start);

}

fclose(fq);

return start;

}

Scnode\* displayScheme(Scnode\* start)

{

cout<<"\nIN DISPLAY\n";

Scnode\* ptr;

ptr = start;

if(ptr == NULL) cout<<"List Empty\n";

else

{

cout<<setw(15)<<"SUBJECT CODE"<<setw(15)<<"SUBJECT TITLE"

<<setw(15)<<"MM SESSIONAL"<<setw(15)<<"MM FINALS"<<setw(15)<<"SEMESTER\n";

while(ptr != NULL)

{

cout<<setw(15)<<ptr->sub\_code<<setw(15)<<ptr->sub\_title<<

setw(15)<<ptr->sess\_marks<<setw(15)<<ptr->theo\_marks<<

setw(15)<<ptr->sem<<endl;

ptr = ptr->next;

}

}

return start;

}

Scnode\* addnode(Scnode\* start, Scnode\* sc)

{

cout<<"\nIN ADDNODE\n";

Scnode\* newnode;

newnode = (Scnode\*)malloc(sizeof(Scnode));

Scnode\* ptr;

strcpy(newnode->sub\_code,sc->sub\_code);

strcpy(newnode->sub\_title,sc->sub\_title);

newnode->sess\_marks = sc->sess\_marks;

newnode->theo\_marks = sc->theo\_marks;

newnode->sem = sc->sem;

if(start == NULL)

{

cout<<"Making the first node\n";

start = newnode;

newnode->next = NULL;

}

else

{

cout<<"Making new node\n";

ptr = start;

while(ptr->next != NULL) ptr = ptr->next;

ptr->next = newnode;

newnode->next = NULL;

}

return start;

}

Scnode\* deleteScheme(Scnode\* start)

{

FILE\* fp;

fp = fopen("Scheme.txt","wb+");

if(start == NULL)

{

cout<<"List is empty\n";

return start;

}

Scnode\* ptr;

Scnode\* preptr;

char dsub\_code[10];

cout<<"Enter the subject code of the subject to be deleted\n";

fflush(stdin);

gets(dsub\_code);

fflush(stdin);

cout<<"Here\n";

ptr = start;

if(strcmp(ptr->sub\_code,dsub\_code) == 0)

{

start = start->next;

free(ptr);

fwrite(start,sizeof(Scnode),1,fp);

return start;

}

else

{

while(strcmp(ptr->sub\_code,dsub\_code) != 0)

{

fwrite(ptr,sizeof(Scnode),1,fp);

preptr = ptr;

ptr = ptr->next;

}

preptr->next = ptr->next;

free(ptr);

while(preptr->next != NULL)

{

fwrite(preptr->next,sizeof(Scnode),1,fp);

}

return start;

}

}

Stnode\* writeStudentRecord(Stnode\* starts)

{

FILE\* fp;

fp = fopen("Student.txt","ab+");

Stnode\* newnode;

newnode = (Stnode\*)malloc(sizeof(Stnode));

Stnode\* ptr;

intdroll\_no;

char dname[20];

char df\_name[20];

char ddob[10];

intdsem;

intdyoa;

cout<<"Enter roll number\n";

cin>>droll\_no;

cout<<"Enter name of the student\n";

fflush(stdin);

gets(dname);

fflush(stdin);

cout<<"Enter Fathers name of the student\n";

fflush(stdin);

gets(df\_name);

fflush(stdin);

cout<<"Enter Date of Birth\n";

cin>>ddob;

cout<<"Enter semester number (1-8)\n";

cin>>dsem;

cout<<"Enter year of admission\n";

cin>>dyoa;

newnode->roll\_no = droll\_no;

strcpy(newnode->name,dname);

strcpy(newnode->f\_name,df\_name);

strcpy(newnode->dob,ddob);

newnode->sem = dsem;

newnode->yoa = dyoa;

if(starts == NULL)

{

cout<<"Making the first node\n";

starts = newnode;

newnode->next = NULL;

cout<<"Writing start to file\n";

fwrite(starts,sizeof(Stnode),1,fp);

}

else

{

cout<<"Making new node\n";

ptr = starts;

while(ptr->next != NULL) ptr = ptr->next;

ptr->next = newnode;

newnode->next = NULL;

cout<<"Writing node to file\n";

fwrite(ptr->next,sizeof(Stnode),1,fp);

}

fclose(fp);

return starts;

}

Stnode\* displayStudentRecord(Stnode\* starts)

{

Stnode\* ptr;

ptr = starts;

if(ptr == NULL) cout<<"List Empty\n";

else

{

cout<<setw(18)<<"ROLL NUMBER"<<setw(20)<<"NAME"<<

setw(20)<<"FATHERS NAME"<<setw(18)<<"DATEOFBIRTH"<<

setw(18)<<"SEMESTER"<<setw(18)<<"ADMISSION YEAR\n";

while(ptr != NULL)

{

cout<<setw(18)<<ptr->roll\_no<<setw(20)<<ptr->name<<

setw(20)<<ptr->f\_name<<setw(18)<<ptr->dob<<

setw(18)<<ptr->sem<<setw(18)<<ptr->yoa<<endl;

ptr = ptr->next;

}

}

return starts;

}

Stnode\* addnodeStudent(Stnode\* starts, Stnode\* st)

{

cout<<"\nIN ADDNODE\n";

Stnode\* newnode;

newnode = (Stnode\*)malloc(sizeof(Stnode));

Stnode\* ptr;

newnode->roll\_no = st->roll\_no;

strcpy(newnode->name,st->name);

strcpy(newnode->f\_name,st->f\_name);

strcpy(newnode->dob,st->dob);

newnode->sem = st->sem;

newnode->yoa = st->yoa;

if(starts == NULL)

{

cout<<"Making the first node\n";

//ptr = start;

starts = newnode;

newnode->next = NULL;

}

else

{

cout<<"Making new node\n";

ptr = starts;

while(ptr->next != NULL) ptr = ptr->next;

ptr->next = newnode;

newnode->next = NULL;

}

return starts;

}

Stnode\* readStudentRecord(Stnode\* starts)

{

FILE\* fq;

fq = fopen("Student.txt","rb+");

Stnodest;

fread(&st,sizeof(Stnode),1,fq);

while(!feof(fq))

{

cout<<"\nYES\n";

starts = addnodeStudent(starts,&st);

cout<<endl<<st.roll\_no<<endl<<st.name<<endl<<st.f\_name<<endl<<st.dob<<endl<<st.sem<<endl<<st.yoa<<endl;

fread(&st,sizeof(Stnode),1,fq);

starts = displayStudentRecord(starts);

}

fclose(fq);

return starts;

}

Rnode\* input\_marks(Rnode\* start)

{

FILE\* fp;

FILE\* ft;

Scnodesc;

Stnodest;

Rnodert;

intsem;

cout<<"Enter semester number\n";

cin>>sem;

intmasess;

intmatheo;

int droll;

fp = fopen("Student.txt","rb+");

ft = fopen("Scheme.txt","rb+");

fread(&st, sizeof(Stnode),1,fp);

fread(&sc,sizeof(Scnode),1,ft);

while(!feof(ft)) //scheme loop

{

rewind(fp);

fread(&st, sizeof(Stnode),1,fp);

while( !feof(fp)) //student loop

{

if(sc.sem == st.sem&&sc.sem == sem)

{

cout<<"In "<<sc.sem<<" semester\n";

cout<<"Enter roll number of the student\n";

cin>>droll;

if(st.roll\_no == droll)

{

cout<<"Enter marks awarded in sessionals\n";

cin>>masess;

cout<<"Enter marks awarded in end semester theory\n";

cin>>matheo;

rt.r\_roll\_no = st.roll\_no;

strcpy(rt.r\_name,st.name);

rt.r\_sem = st.sem;

rt.ma\_sess = masess;

rt.ma\_theo = matheo;

startl = writeRecord(startl, &rt);

}

}

fread(&st,sizeof(Stnode),1,fp);

}

fread(&sc,sizeof(Scnode),1,ft);

}

fclose(fp);

fclose(ft);

return startl;

}

Rnode\* writeRecord(Rnode\* startl, Rnode\* rt)

{

FILE\* fs;

fs = fopen("Record.txt","ab+");

Rnode\* newnode;

newnode = (Rnode\*)malloc(sizeof(Rnode));

Rnode\* ptr;

newnode->r\_roll\_no = rt->r\_roll\_no;

strcpy(newnode->r\_name,rt->r\_name);

newnode->r\_sem = rt->r\_sem;

newnode->ma\_sess = rt->ma\_sess;

newnode->ma\_theo = rt->ma\_theo;

if(startl == NULL)

{

cout<<"Making the first node\n";

startl = newnode;

newnode->next = NULL;

cout<<"Writing startl to file\n";

fwrite(startl,sizeof(Rnode),1,fs);

}

else

{

cout<<"Making new node\n";

ptr = startl;

while(ptr->next != NULL) ptr = ptr->next;

ptr->next = newnode;

newnode->next = NULL;

cout<<"Writing node to file\n";

fwrite(ptr->next,sizeof(Rnode),1,fs);

}

fclose(fs);

return startl;

}

Rnode\* readRecord(Rnode\* startl)

{

FILE\* fq;

fq = fopen("Record.txt","rb+");

Rnodert;

fread(&rt,sizeof(Rnode),1,fq);

while(!feof(fq))

{

cout<<"\nYES\n";

startl = addnodeRecord(startl,&rt);

cout<<endl<<rt.r\_roll\_no<<endl<<rt.r\_name<<endl<<rt.r\_sem<<endl<<rt.ma\_sess<<endl<<rt.ma\_theo<<endl;

fread(&rt,sizeof(Rnode),1,fq);

startl = displayMarksRecord(startl);

}

fclose(fq);

return startl;

}

Rnode\* addnodeRecord(Rnode\* startl, Rnode\* rt)

{

cout<<"\nIN ADDNODE\n";

Rnode\* newnode;

newnode = (Rnode\*)malloc(sizeof(Rnode));

Rnode\* ptr;

newnode->r\_roll\_no = rt->r\_roll\_no;

cout<<"Roll number assigned to new node is "<<rt->r\_roll\_no<<endl;

strcpy(newnode->r\_name,rt->r\_name);

newnode->r\_sem = rt->r\_sem;

newnode->ma\_sess = rt->ma\_sess;

newnode->ma\_theo = rt->ma\_theo;

if(startl == NULL)

{

cout<<"Making the first node\n";

startl = newnode;

newnode->next = NULL;

}

else

{

cout<<"Making new node\n";

ptr = startl;

while(ptr->next != NULL) ptr = ptr->next;

ptr->next = newnode;

newnode->next = NULL;

}

return startl;

}

Rnode\* displayMarksRecord(Rnode\* startl)

{

Rnode\* ptr;

ptr = startl;

if(ptr == NULL) cout<<"List Empty\n";

else

{

cout<<setw(18)<<"ROLL NUMBER"<<setw(20)<<"NAME"<<

setw(18)<<"SEMESTER"<<setw(18)<<"SESSIONAL MARKS"<<

setw(18)<<"FINALS MARKS"<<endl;

while(ptr != NULL) //not ptr->next, it should be ptr != NULL

{

cout<<setw(18)<<ptr->r\_roll\_no<<setw(20)<<ptr->r\_name<<

setw(18)<<ptr->r\_sem<<setw(18)<<ptr->ma\_sess<<

setw(18)<<ptr->ma\_theo<<endl;

ptr = ptr->next;

}

}

return startl;

}

Rnode\* printRecordSemester(Rnode\* startl)

{

Rnode\* ptr;

intdsem;

cout<<"Enter semester number\n";

cin>>dsem;

ptr = startl;

if(ptr == NULL) cout<<"List Empty\n";

else

{

cout<<"IN "<<dsem<<" SEMESTER\n\n";

cout<<setw(18)<<"ROLL NUMBER"<<setw(20)<<"NAME"<<

setw(18)<<"SEMESTER"<<setw(18)<<"SESSIONAL MARKS"<<

setw(18)<<"FINALS MARKS"<<endl;

while(ptr != NULL)

{

if(ptr->r\_sem == dsem)

{

cout<<setw(18)<<ptr->r\_roll\_no<<setw(20)<<ptr->r\_name<<

setw(18)<<ptr->r\_sem<<setw(18)<<ptr->ma\_sess<<

setw(18)<<ptr->ma\_theo<<endl;

}

ptr = ptr->next;

}

}

return startl;

}

Rnode\* listRecord(Rnode\* startl)

{

Rnode\* ptr;

intdsem;

cout<<"Enter semester number\n";

cin>>dsem;

int droll;

cout<<"Enter roll number to access\n";

cin>>droll;

int flag = 0;

ptr = startl;

if(ptr == NULL) cout<<"List Empty\n";

else

{

cout<<"IN "<<dsem<<" SEMESTER\n\n";

cout<<setw(18)<<"ROLL NUMBER"<<setw(20)<<"NAME"<<

setw(18)<<"SEMESTER"<<setw(18)<<"SESSIONAL MARKS"<<

setw(18)<<"FINALS MARKS"<<endl;

while(ptr != NULL)

{

if(ptr->r\_sem == dsem&&ptr->r\_roll\_no == droll)

{

flag = 1;

cout<<setw(18)<<ptr->r\_roll\_no<<setw(20)<<ptr->r\_name<<

setw(18)<<ptr->r\_sem<<setw(18)<<ptr->ma\_sess<<

setw(18)<<ptr->ma\_theo<<endl;

}

ptr = ptr->next;

}

if(flag == 0) cout<<"The given record does not exist\n";

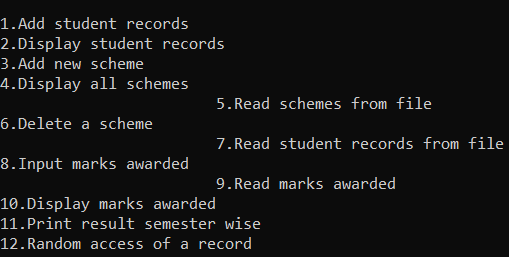
}

return startl;

}

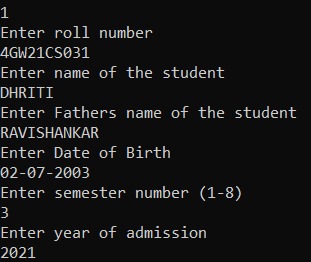
**SAMPLE INPUTS**

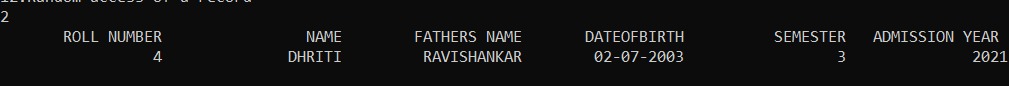
Input screen



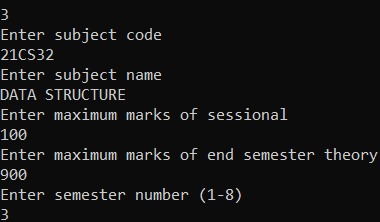
**SAMPLE OUTPUT**

Case1:

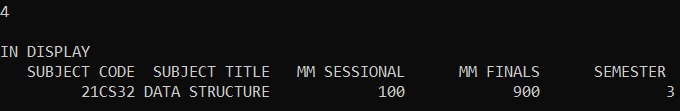


Case 2: ****

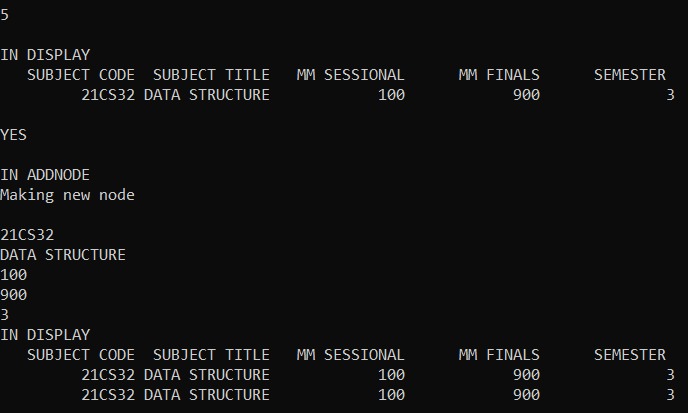
Case 3:

****

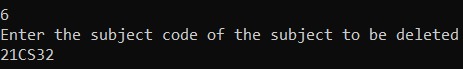
Case 4:



Case 5:



Case 6:



**CONCLUSIONS AND FUTURE SCOPE**

**CONCLUSION**

In this project, we used the programming language (C++) in such a way that it can store and process student data, scores, and schemes that execute results semester-wise. This helps the faculties to analyse their student’s results and pass these reports to other members of the institution. This allows reports to be more easily referenced for future comparisons of students’ progress and it allows faculties the space for other areas of work as a single database store all mark sheets. Updating information becomes easier. This will avoid calculation and simplify the process of visualizing results by students as well as faculty. It is concluded that the system will works well and thus it will fulfil the user's requirement.

**FUTURE SCOPE**

The future scope is that data can be fetched, parsed in other formats like doc, csv, odt, etc. Visualization can be provided to represent data in graphical format various representation like pie chart, graph, etc. Enhancements can be done to maintain all the student result, subject, class and semester.

**REFERENCES**

**WEBSITES**

* [www.geeksforgeeks.com](http://www.geeksforgeeks.com)
* [www.w3schools.com](http://www.w3schools.com)
* [www.github.com](http://www.github.com)
* [www.codeproject.com](http://www.codeproject.com)
* [www.sanfoundry.com](http://www.sanfoundry.com)
* [www.reasearchgate.com](http://www.reasearchgate.com)
* [www.codewithc.com](http://www.codewithc.com)
* [www.log2base.com](http://www.log2base.com)
* [www.tutorialpoint.com](http://www.tutorialpoint.com)
* [www.en.m.wikipedia.com](http://www.en.m.wikipedia.com)
* [www.google.com](http://www.google.com)