SRM Student Admission Management System

Project submitted to the

SRM University - AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering School of Engineering and Sciences

Submitted by

Vamsi Kumar A - (AP21110010135)

Manoj Kumar G - (AP21110010147)

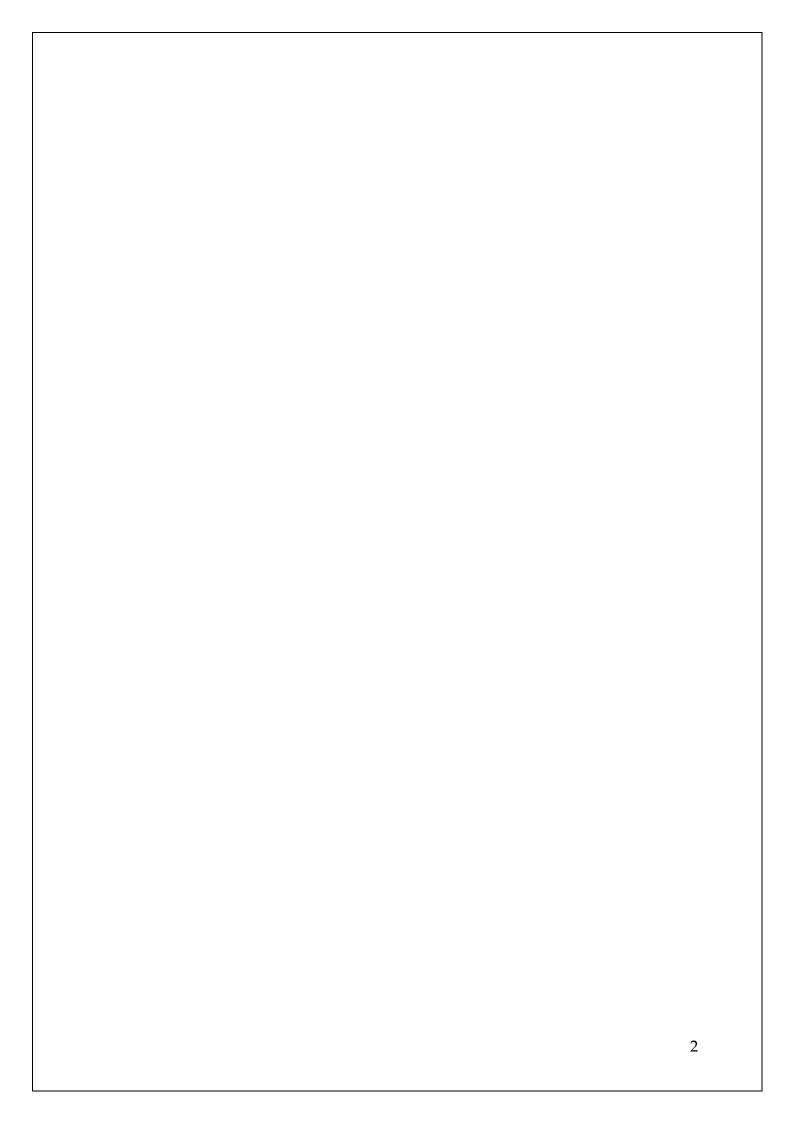
Teja P - (AP21110010155)

Susritha V - (AP21110010185)



Under the Guidance of (RAJIV SENAPATI)

SRM University-AP
Neerukonda, Mangalagiri, Guntur
Andhra Pradesh - 522 240
December, 2022



Certificate

Date: 11-Dec-22

This is to certify that the work present in this Project entitled "SRM STUDENT ADMISSION MANAGEMENT SYSTEM" has been carried out by Group 11(vamsi kumar, manoj kumar, Teja, Susritha) my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in School of Engineering and Sciences.

Supervisor

(Signature)

Dr. Rajiv Senapati

Assistant Professor, CSE Department

SRM UNIVERSITY, AP.

Co-supervisor

(Signature)

Venkaiah Chowdary B

LAB Assistant,

SRM UNIVERSITY, AP.

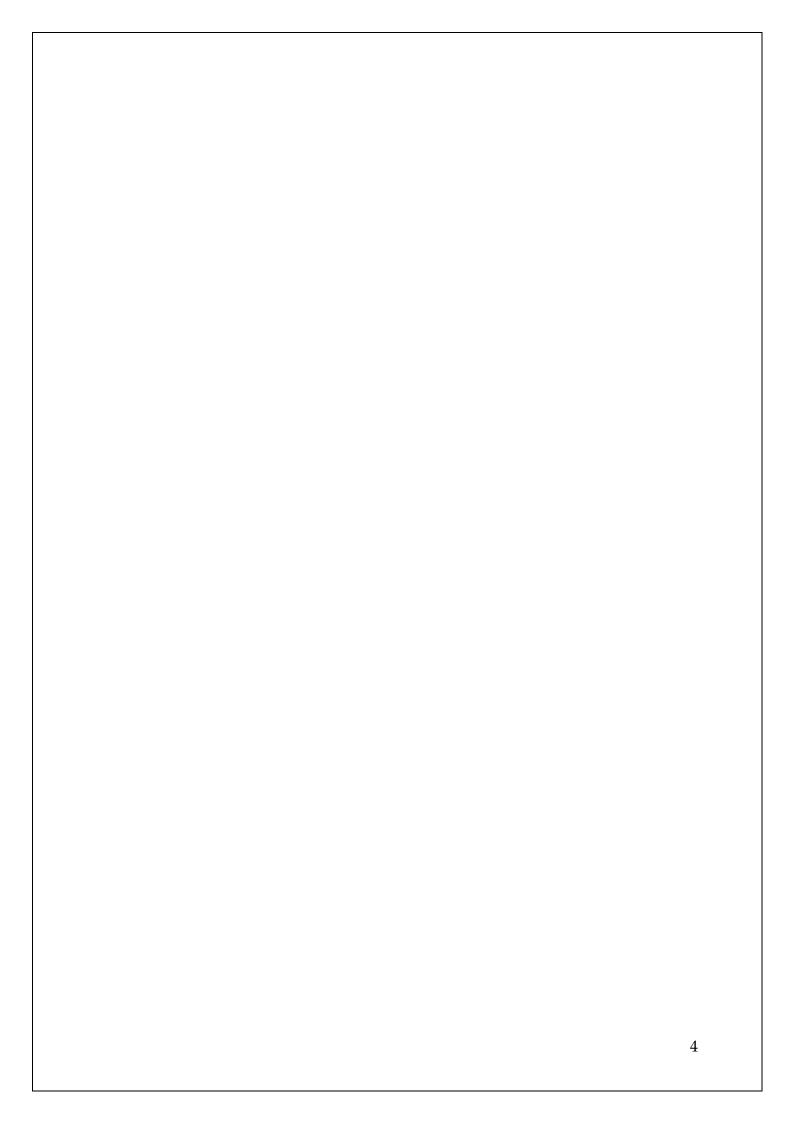
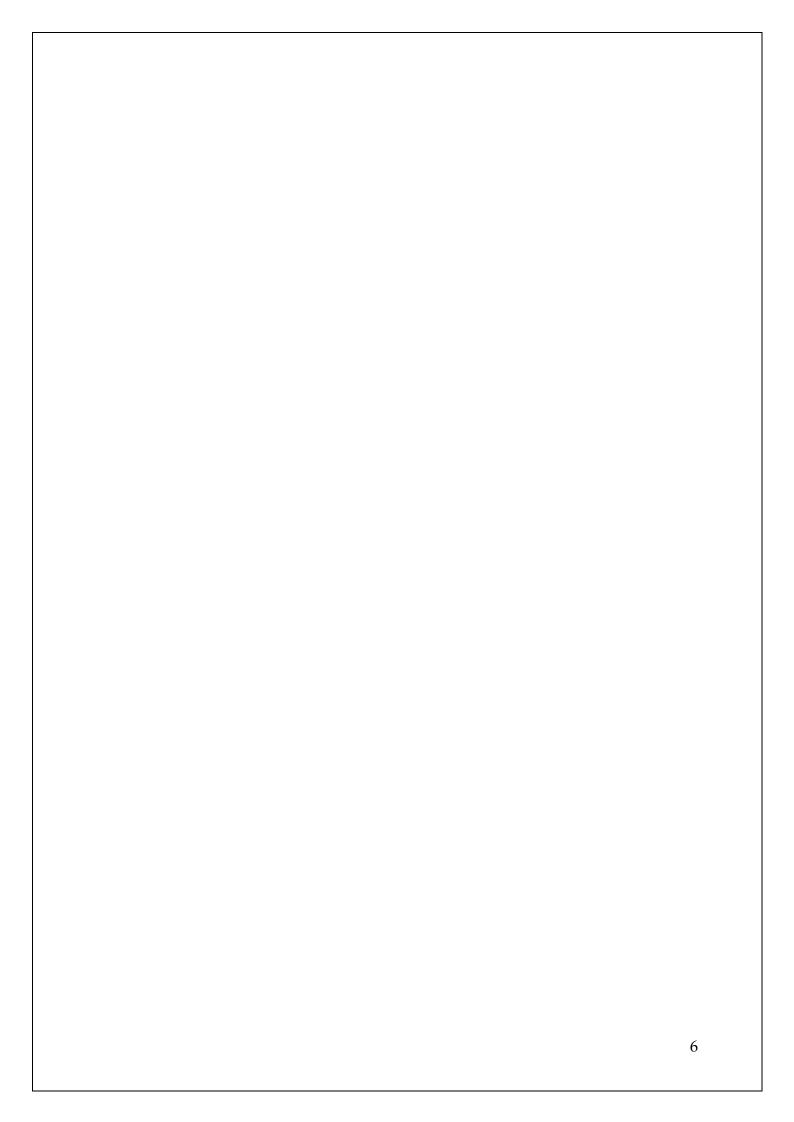


TABLE OF CONTENTS

Certif	icate	3	
Table	of content	rs5	
Abstr	act	7	
1.	Introduc	iction9	
2.	Methodology		
	2.1 Imple	ementation	
	2.2 Code		
3. Discussion		on21	
	3.1 Outp	ut	
	3.1.1	Home screen or menu	
	3.1.2	Creating a new student record	
	3.1.3	Search student record	
	3.1.4	Displaying all student records	
	3.1.5	Delete student record	
	3.1.6	Modify student record	
	3.1.7	Exiting the program.	
4	Conclusion	on27	



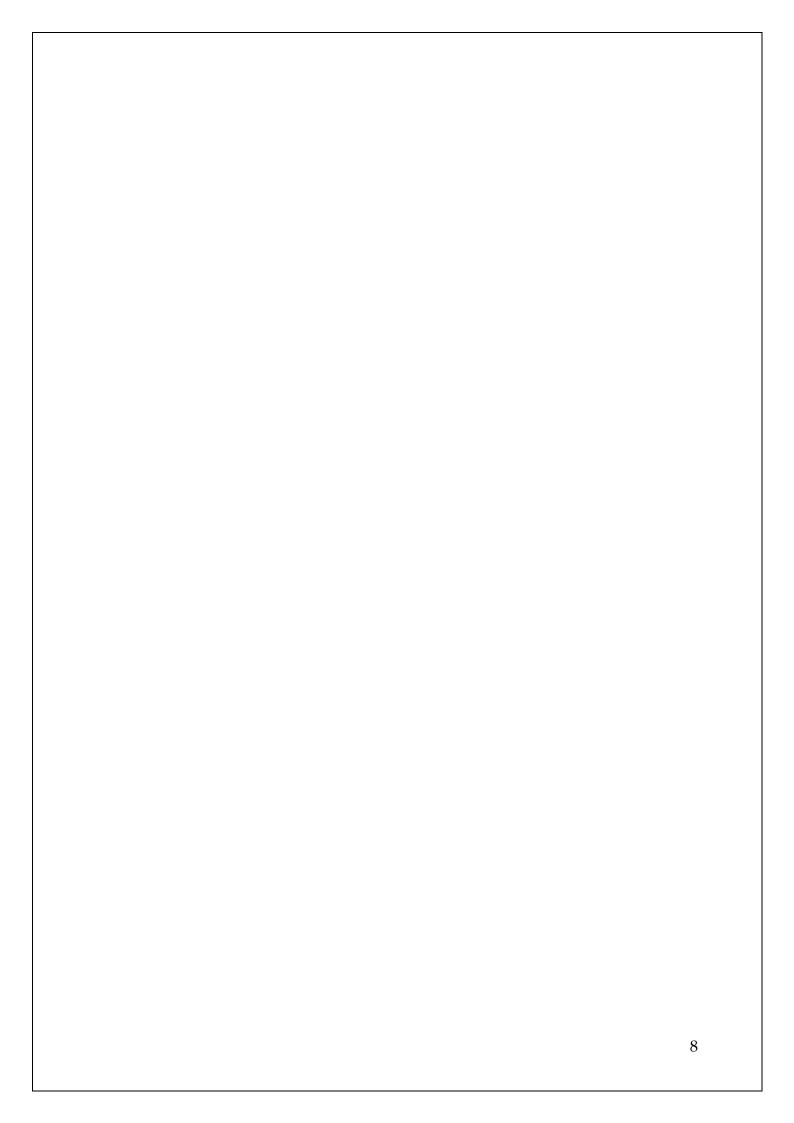
Abstract:

Today all the work at the time of admission of the students is done manually by ink and paper, which is very slow and consuming much efforts and time. It is required to Design of a Computerized Automated Student Admission System, to speed up and make it easy to use system.

Student admissions are a vital part of any university's running. The process begins with a potential student completing an application form through the Universities and Colleges Admissions Service, the first step for students is to apply directly to the university through a custom online form.

This project's aim is to automate the system, prechecking the inclusion of all required material and automatically ranking each student's application based on a number of criteria.

These criteria include the ranking of their university, their grade at said university and their language grade Certificate. The data used by the system is stored in files that will be the center of all information held about students and the base for the remainder of the process after the initial application has been made. This enables things to be simplified and considerably quickened, making the jobs of the people involved easier.



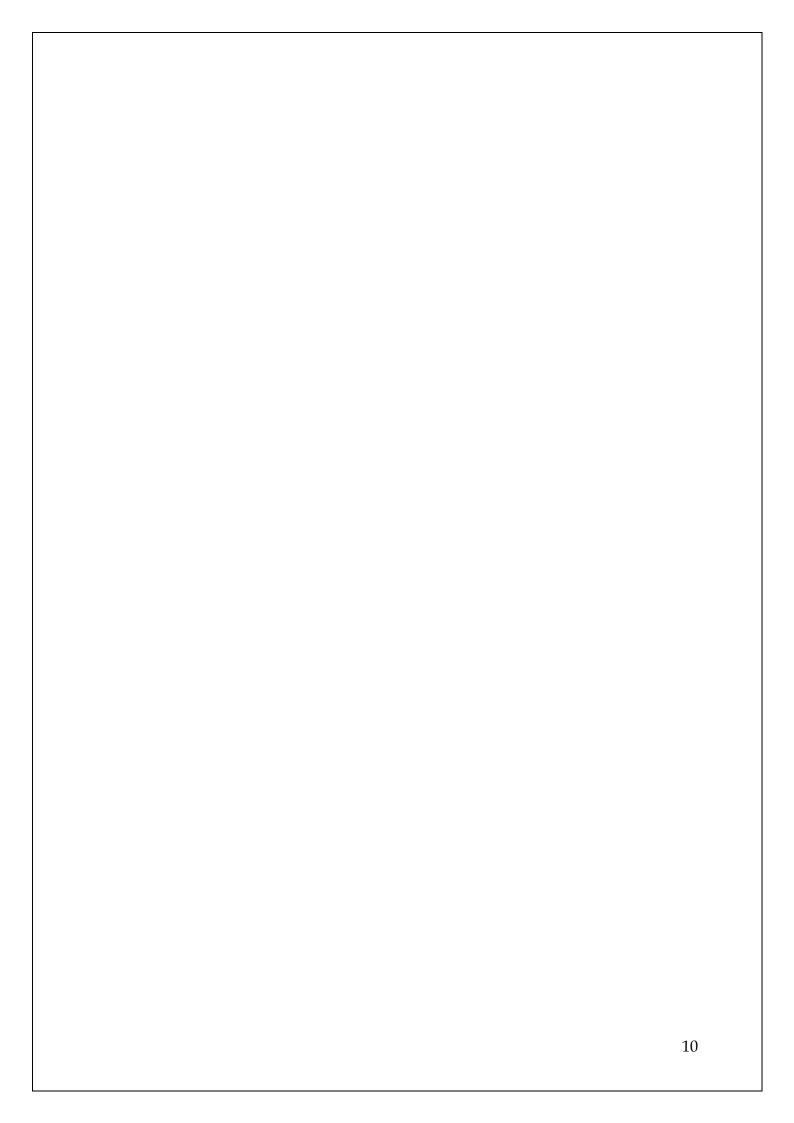
Introduction

We are going to present Student admission management system. We use records or books to store details of the student and store them.

But data may get lost when we lose the book. So, we are going to design a code where we store the data of the students and we store them in a file so that we can access in a min and search them in an instant.

We designed this code in such a way that it stores the data of every student and we can access it when we want.

In this code we will take the input from the student and store them in a file. So, that if we need details of the student we can use it for further details. We will use the roll.no to find the student and we can print every student at a time. The file we use will be saved in our project folder and we can have them in a folder and access them.



METHODOLOGY

At first we need a way to represent all the information and to extract useful information from this.

We used the #include<windows.h> to integrate our program with the windows and to use the command prompt screen to display content of our program ,

To build a real student with all the characteristics, we build a class as the name with "Student" with components like,

Name,

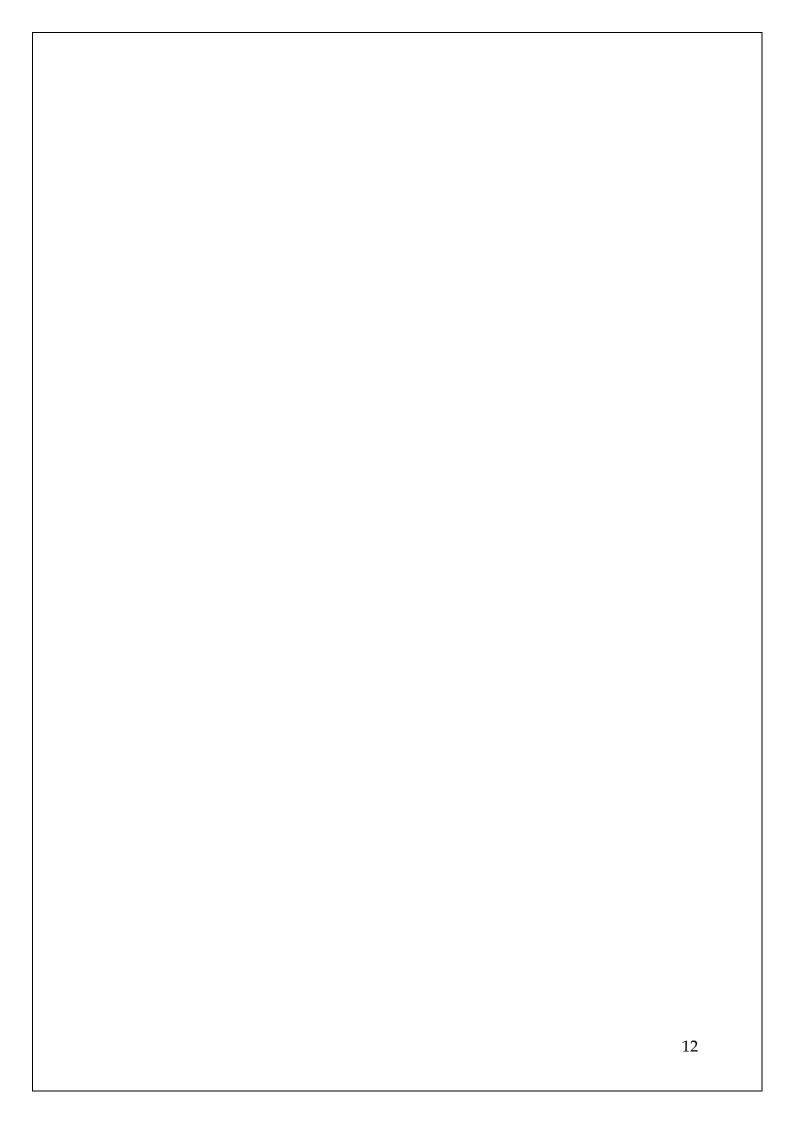
Roll no,

Subject marks,

Average,

Grade,

Fee.



IMPLEMENTATION:

CODE:

```
#include<iostream>
 2
    #include<fstream>
3
    #include<iomanip>
    #include<stdlib.h>
    using namespace std;
     // the class that stores data
8
     class student
9
10
         int rollno;
         char name[50];
11
         int eng_marks, math_marks, sci_marks, lang2_marks, cs_marks;
12
13
         double average;
14
         char grade;
15
        int fee;
16
17
         public:
18
         void getdata();
19
         void showdata() const;
20
         void calculate();
21
         int retrollno() const;
22
    };
23
    //class ends here
24
25
     void student::calculate()
26
27
         average=(eng_marks+math_marks+sci_marks+lang2_marks+cs_marks)/5.0;
         if(average>=90){
28
29
         grade='A';
30
31
         fee=20000;}
32
         else if(average>=75){
33
34
         grade='B';
35
         fee=50000;}
36
         else if(average>=50){
```

```
37
          grade='C';
38
          fee=75000;}
39
40
          else{
41
          grade='F';
42
43
          fee=100000;
44
45
46
47
      void student::getdata()
48
          cout<<"Enter student's roll number: ";</pre>
49
50
          cin>>rollno;
51
          cout<<"Enter student's name: ";</pre>
52
53
          cin.ignore();
54
          cin.getline(name,50);
55
56
          cout<<"\n\nEnter Student's Marks"<<endl;</pre>
          cout<<"All marks should be out of 100";</pre>
57
58
          cout<<"\n\nEnter marks in English: ";</pre>
59
60
          cin>>eng_marks;
61
62
          cout<<"Enter marks in Math: ";</pre>
63
          cin>>math_marks;
64
65
          cout<<"Enter marks in Science: ";</pre>
66
          cin>>sci_marks;
67
          cout<<"Enter marks in Second language: ";</pre>
68
          cin>>lang2_marks;
69
70
          cout<<"Enter marks in Social Studies: ";</pre>
71
```

```
72
           cin>>cs_marks;
 73
           calculate();
 74
 75
      void student::showdata() const
 76
 77
           cout<<"\nRoll number of student : "<<rollno;</pre>
 78
           cout<<"\nName of student : "<<name;</pre>
 79
           cout<<"\nEnglish : "<<eng_marks;</pre>
 80
           cout<<"\nMaths : "<<math_marks;</pre>
 81
           cout<<"\nScience : "<<sci_marks;</pre>
 82
           cout<<"\nSecond Language : "<<lang2_marks;</pre>
 83
           cout<<"\nComputer Science :"<<cs_marks;</pre>
 84
           cout<<"\nAverage Marks :"<<average;</pre>
 85
           cout<<"\nGrade of student is :"<<grade;</pre>
 86
           cout<<"\nFee of the student is :"<<fee;</pre>
 87
 88
 89
 90
 91
      int student::retrollno() const
 92
 93
      return rollno;
 94
 95
 96
      //function declaration
97
98
      void create student();
      void display_sp(int);//display particular record
99
      void display_all(); // display all records
100
      void delete_student(int);//delete particular record
101
      void change_student(int);//edit particular record
102
103
104
      //Write Student Details to File
105
106
      void create_student()
107
      {
```

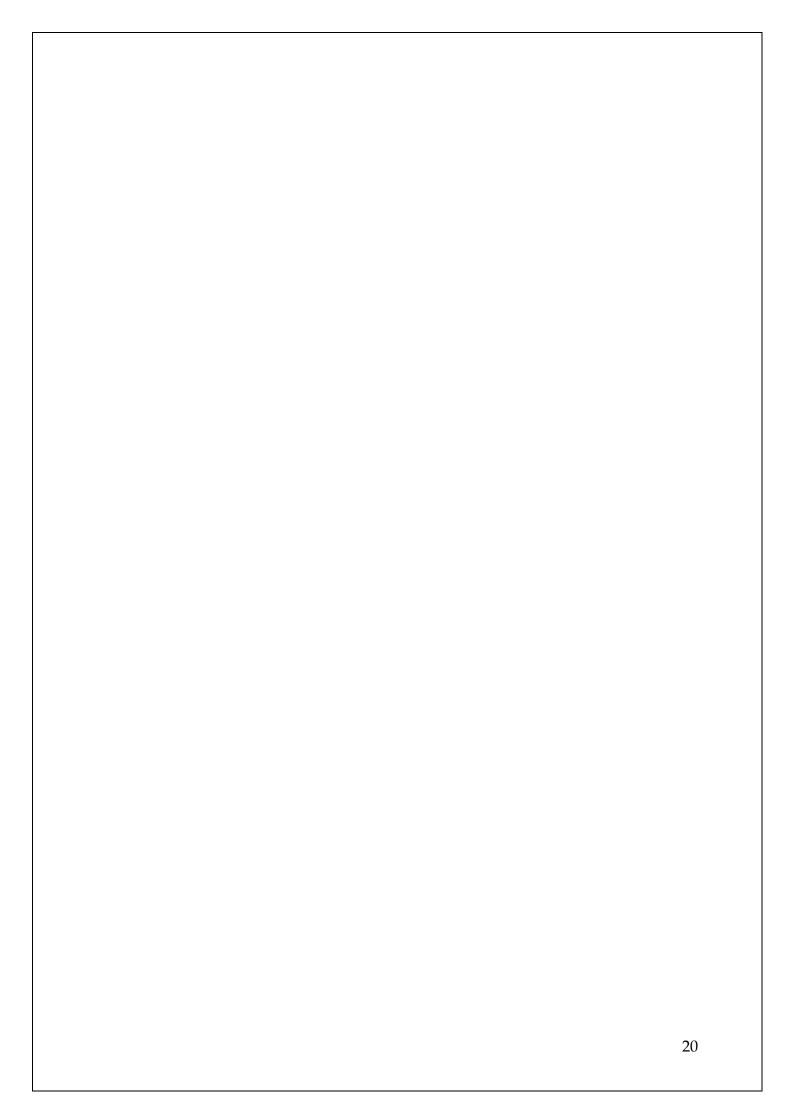
```
108
          student stud;
          ofstream oFile;//it is used to create files and write information to the files
109
          oFile.open("SSS.dat",ios::binary|ios::app);
110
111
          stud.getdata();
          oFile.write(reinterpret_cast<char *> (&stud), sizeof(student));
112
          oFile.close();
113
          cout<<"\nStudent's record has been created ";</pre>
114
          cin.ignore();
115
116
          cin.get();
117
118
      // Read File Records
119
120
      void display_all()
121
122
123
          student stud;
124
          ifstream inFile;//read information from files
125
          inFile.open("SSS.dat",ios::binary);
126
          if(!inFile)
127
128
          cout<<"File could not be opened !! Press any Key to exit";</pre>
129
          cin.ignore();
130
          cin.get();
131
          return;
132
133
      cout<<"\n\n\t\tDISPLAYING ALL RECORDS\n\n";</pre>
134
      while(inFile.read(reinterpret_cast<char *> (&stud), sizeof(student)))
135
136
          stud.showdata();
137
          cout<<"\n\n======\n";
138
139
140
141
      inFile.close();
142
      cin.ignore();
143
      cin.get();
```

```
144
      }
145
146
      //Read Specific Record Based on Roll Number
147
      void display_sp(int n)
148
149
150
          student stud;
          ifstream iFile;
151
          iFile.open("SSS.dat",ios::binary);
152
          if(!iFile)
153
154
155
              cout<<"File could not be opened... Press any Key to exit";</pre>
156
157
              cin.ignore();
158
              cin.get();
159
              return;
160
161
          bool flag=false;
162
          while(iFile.read(reinterpret_cast<char *> (&stud), sizeof(student)))
163
164
              if(stud.retrollno()==n)
165
166
                   stud.showdata();
167
168
                   flag=true;
169
170
171
          iFile.close();
172
173
          if(flag==false)
174
          cout<<"\n\nrecord does not exist";</pre>
175
          cin.ignore();
          cin.get();
176
177
```

```
// Modify Record for Specified Roll Number
179
180
181
      void change_student(int n)
182
183
          bool found=false;
184
          student stud:
          fstream fl;//it is used to create files write information to files and read information from files
185
186
          fl.open("SSS.dat",ios::binary|ios::in|ios::out);
          if(!fl)
187
188
189
190
              cout<<"File could not be opened. Press any Key to exit...";</pre>
191
              cin.ignore();
192
              cin.get();
193
              return:
194
195
          while(!fl.eof() && found==false)
196
197
198
               fl.read(reinterpret_cast<char *> (&stud), sizeof(student));
199
              if(stud.retrollno()==n)
200
201
                   stud.showdata();
                  cout<<"\nEnter new student details:"<<endl;</pre>
202
203
                   stud.getdata();
204
                   int pos=(-1)*static_cast<int>(sizeof(stud));
205
                   fl.seekp(pos,ios::cur);
                   fl.write(reinterpret_cast<char *> (&stud), sizeof(student));
206
207
                   cout<<"\n\n\t Record Updated";</pre>
208
                   found=true;
209
210
211
          fl.close();
212
213
          if(found==false)
214
           cout<<"\n\n Record Not Found ";</pre>
           cin.ignore();
215
216
           cin.get();
217
218
219
      //Delete Record with Particular Roll Number
220
221
      void delete_student(int n)
222
223
           student stud;
           ifstream iFile;//read the information from files
224
           iFile.open("SSS.dat",ios::binary);
225
226
           if(!iFile)
227
228
               cout<<"File could not be opened... Press any Key to exit...";</pre>
229
               cin.ignore();
230
               cin.get();
231
               return;
232
233
           ofstream oFile;//it is used to create files and write information to the files
           oFile.open("Temp.dat",ios::out);
234
235
           iFile.seekg(0,ios::beg);
236
           while(iFile.read(reinterpret cast<char *> (&stud), sizeof(student)))
237
               if(stud.retrollno()!=n)
238
239
240
                    oFile.write(reinterpret_cast<char *> (&stud), sizeof(student));
241
242
243
           oFile.close();
244
           iFile.close();
245
           remove("SSS.dat");
           rename("Temp.dat","SSS.dat");
246
           cout<<"\n\n\tRecord Deleted ..";</pre>
247
248
           cin.ignore();
249
           cin.get();
```

178

```
250
251
      // main function
252
253
254
      int main()
255
           char ch;
256
257
           cout<<setprecision(2);</pre>
258
259
260
               char ch;
261
               int num;
262
               system("cls");//clearing the screen
263
264
               cout<<"Welcome to Student Admission Management System"<<endl;</pre>
               cout<<"\nMENU"<<endl;</pre>
265
               cout<<"\n1. Create a newstudent record"<<endl;</pre>
266
267
               cout<<"2. Search student record"<<endl;</pre>
268
               cout<<"3. Display all students records"<<endl;</pre>
               cout<<"4. Delete student record"<<endl;</pre>
269
               cout<<"5. Modify student record"<<endl;</pre>
270
               cout<<"6. Exit"<<endl;</pre>
271
272
               cout<<"\nEnter your choice"<<endl;</pre>
273
               cin>>ch;
274
               system("cls");
275
276
277
           switch(ch)
278
               case '1': create_student(); break;
279
               case '2': cout<<"\n\n\tEnter The roll number: ";</pre>
280
281
               cin>>num;
282
               display_sp(num); break;
               case '3': display_all(); break;
283
               case '4':
                              cout<<"\n\n\tEnter The roll number: ";</pre>
284
285
               cin>>num;
                 delete_student(num);break;
286
287
                 case '5':
                                  cout<<"\n\n\tEnter The roll number: "; cin>>num;
288
                 change student(num);break;
                 case '6':
                                 cout<<"Exiting, Thank you!";exit(0);</pre>
289
290
291
            }while(ch!='6');
292
293
            return 0;
294
```



DISCUSSION:

OUTPUT

1) HOME SCREEN (OR) MENU

```
Welcome to Student Admission Management System

MENU

1. Create a newstudent record
2. Search student record
3. Display all students records
4. Delete student record
5. Modify student record
6. Exit

Enter your choice
```

2) CREATING A NEW STUDENT RECORD

```
Enter student's roll number: 1
Enter Student's Marks
All marks should be out of 100

Enter marks in English: 89
Enter marks in Math: 89
Enter marks in Science: 89
Enter marks in Second language: 90
Enter marks in Social Studies: 98

Student's record has been created
```

3) SEARCH STUDENT RECORD

Enter The roll number: 1

Roll number of student : 1

Name of student : VAMSI

English : 89

Maths : 89

Science : 89

Second Language : 90

Computer Science :98

Average Marks :91

Grade of student is :A

Fee of the student is :20000

4) DISPLAYING ALL STUDENT RECORDS

DISPLAYING ALL RECORDS Roll number of student : 1 Name of student : VAMSI English: 89 Maths: 89 Science: 89 Second Language: 90 Computer Science :98 Average Marks :91 Grade of student is :A Fee of the student is :20000 _____ Roll number of student : 2 Name of student : manoj English : 78 Maths: 98 Science: 78 Second Language: 67 Computer Science :98 Average Marks :84 Grade of student is :B Fee of the student is :50000 ______

5) DELETE STUDENT RECORD

```
Enter The roll number: 1
Record Deleted ..
```

6) MODIFY STUDENT RECORD

```
Enter The roll number: 2
Roll number of student : 2
Name of student : manoj
English: 78
Maths: 98
Science: 78
Second Language: 67
Computer Science :98
Average Marks :84
Grade of student is :B
Fee of the student is :50000
Enter new student details:
Enter student's roll number: 1
Enter student's name: vamsi
Enter Student's Marks
All marks should be out of 100
Enter marks in English: 98
Enter marks in Math: 78
Enter marks in Science: 98
Enter marks in Second language: 89
Enter marks in Social Studies: 89
         Record Updated
```

7) EXITING THE PROGRAM

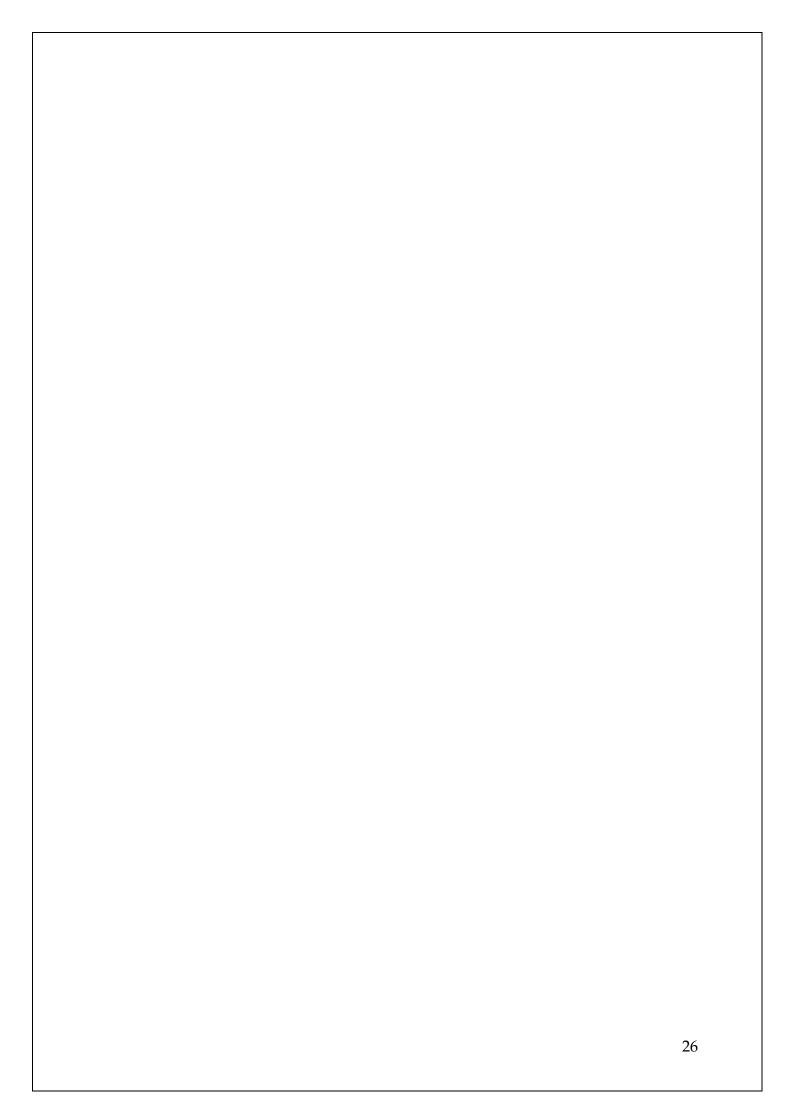
➤ After selecting the 6 it will exit the program

```
Welcome to Student Admission Management System

MENU

1. Create a newstudent record
2. Search student record
3. Display all students records
4. Delete student record
5. Modify student record
6. Exit

Enter your choice
6
```



Conclusion:

This code is used to accept the student details and previous year marks of the student which we can use to store information. By using the above details we can say them how much they want to pay for upcoming year. We can store data in this code and we can just rank them and access the code whenever required but if even if we get an error while entering the details we corrupt the information or file which we are storing the data.

While writing this program we focused on making the code simple and easy to understand. The code should be user-friendly and easy to understand. The interface which we designed has details of student which should be entered and we can search the student by entering the roll no.

This is basic student admission management system code.

