**A Project Report**

**STUDENT RECORD SYSTEMS**

**Mini-project lab**

**By**

**MOHAMMAD ABDUL IMRAN (17R01A05E7)**

**MOHD SUBHAN (17R01A05E9)**

**MANDALA SHIVA KUMAR (18R05A0525)**

**SAARA SANTOSH GOUD (18R05A0535)**

**Department of Computer Science and Engineering**

**CMR INSTITUTE OF TECHNOLOGY**

**Hyderabad – 501 401**

**CONTENTS**

**Acknowledgements**

**Abstract**

**List of Figures**

**Introduction**

* History
* Objective of the Project
* Limitations of the Project
* Problem

**Literature Survey**

* Introduction
* Existing System
* Proposed System
* Strategies

**Analysis**

* + Introduction
  + Advantages and disadvantages
  + Software & hardware Requirements
  + Flowchart
  + How it works ?

**Implementation and Design**

* + Introduction
  + Front-End
  + Back-End
  + Design and Flow of the Graphical User Interface
  + Testing and Validation
    - * Introduction
      * Design of Test Cases and Scenarios
      * Validation
      * Conclusion

**Conclusion and Future Enhancements**

**References**

**Appendix - I**

**PROJECT ON STUDENT RECORD SYSTEMS**

**ABSTRACT**

This project is about Student Record Systems Project. **STUDENT RECORD SYSTEMS** is a Application most oftenly Used by Students And Teachers. Student Record Systems often used as information tool for knowing and analysing the Student Details and credits and it is very useful to retrieve the Student information whenever needed.

Many Applications share the element of trying to be the first to get n-in-a-row, including Many Applications like student analyser and student Reports. STUDENT RECORD SYSTEM S is an instance of Other applications, where everyone can access the Student info by login credentials.

The languages used in this project is c and c++ languages.

By,

MOHAMMAD ABDUL IMRAN (17R01A05E7)

MOHD SUBHAN (17R01A05E9)

MANDALA SHIVA KUMAR (18R05A0525)

SAARA SANTOSH GOUD (18R05A0535)

**Introduction**

* Student Record System (SRS) is a web-based application software designed to introduce a conducive and structured information exchange environment for integrating students, parents, teachers and the administration of a school or college. Some of the other software packages available for this purpose include Student Management System (SMS), Student Information Management System (SIMS) and Student Information System (SIS). These software systems enable educational institutions to supervise student-related activities such as keeping records of tests or examinations conducted, attendance, appraisal on performance including details of marks scored, particulars of everyday school attendance, and all other institution-related activities; in short, they provide a complete student records system. They are designed with diverse application potentials ranging from simple management of students’ records at school to management of all student-related functions as well as administrative functions of a university or a chain of educational establishments.
* **History:-**

Much of the software used in erstwhile student information systems and school administration worked on an outdated code base and had a curious mix of presentation, business sense, and data accessibility. It was impossible and often expensive to modify these systems to cater to future demands.

Some of the early initiative software programs such as WebCT, SCT Campus Pipeline, Jetspeed, and Blackboard merely enabled interactions between students and teachers for academic purposes and served only as a student information system. The advantage of having a student management system was missing as it was not possible to integrate any of the campus-based activities within these systems. Most often, student data were kept strewn in multiple locations, making it difficult to construe and project.

Educational institutions were looking for a web-based solution such as a student information management system or an intelligent student information system with an inbuilt mechanism to meet the requirements of student-related information management as well as to serve the purpose of a school management system. Furthermore, they were looking forward to a complete package on student information management system from a single vendor.

**Objective of the project:-** Student information systems must meet the following objectives:

·Enable self-service for students to perform basic administrative functions and tasks in a “one-stop” service and access environment.

·Integrate data sources and process them through a single function that supports one-time entry of student data.

·Support the retention and recruitment of students and encourage a strong and positive relationship with the university, alumni, donors, and other constituencies.

·Integrate and support new learning and teaching opportunities and technologies for students and faculty.

·Support open interfaces and integration with other campus applications and database systems.

·Ensure data integrity, privacy, and security in an open-access environment.

·Support faculty and staff who perform both basic and complex functions through simplified work processes and procedures.

·Enable ongoing, flexible reconfiguration of the application to extend its life cycle and to meet the changing business processes of the university.

·Include implementation and support services that meet university needs.

**LITERATURE SURVEY:-**

* **Introduction:-**

System has come up with many functionalities for educational institutions to track the student progress and

managing attendance. It helps both student and guardian to keep track of student progress without visiting to the college. It

also notifies student and guardian during the time of important events which are happening in institution. One more feature

is guardian get alerted whenever student get failed in the exam or student not able to meet the expected attendance average.

Student Information Report System (SIRS) is application software and which has intention to begin a conductive and direct

interchanging the statistics in a secure platform to coalesce with students, faculties, parents and the college/school

administration. The student information has the particulars (like register number sem, date-of-birth, sex, parent phone

number, address, parent name, etc.) invade to the system by the faculties. All these particulars is stored in the database.

SIRS application is trouble free to use in schools, colleges, universities, and any other educational institutions. It can be

customized as per the need. It can be used in private and government educational institutions also. SIRS application is an

internet based application we can login to the system from anywhere irrespective of geographical area it will give seamless

navigation. The paper [2] provides the particulars to carry out the performance, management and decision-making

functions of enterprises or organizations. Enormous grow of students is caused to expand the functionality in the respective

educational institutions. As student added to the educational system it is difficult to manage and track student details. To

overcome difficulties we come up with this new approach student information management system with additional features.

This new approach will provide fast processing, efficient student tracking, and produces desired result. This approach will

allow students to save their personal details. It is more secure, reliable and easy to use.

Attendance [3] is part of any system to keep track of the particular person. It is mandatory process in educational system

which directly reflects the student progress. In educational institutes attendance management is normally a manual process.

There is enormous grow in the software industry which has privileged colleges to maintain the attendance system by using

gadgets which is the best way. As we are using the smart phones we not require maintaining attendance register. It can be

easily done in mobile application. Faculty will be going to take the attendance when class gets started. They will initially

login to the system through mobile application. Once attendance has been taken successfully for the class it will sent to

sever through GPRS. The faculties can also do the necessary functions like registering new students, deleting the

information about a particular student, modifying the information regarding the student etc. The main intention of this

process is to reduce the risk of manual efforts. It also reduces the time consumption. Also we want to give importance to

reduce the paper wastage that daily happens. The model [4] utilizes computer aided system. The model plays main role in

an institution or in the college management. Initially, the system has developed with four layers based on the hierarchy

such as Web display layer where application is deployed and displayed for end users. Business logic layer responsible for

handling the functionality of the product. Data access layer is responsible for viewing the data. Database layer responsible

for storing the student data. In Database layer ER diagram has been designed to provide data normalization. The process

provides complete information about student, faculties and educational institution. Third thing in this project is to allowing

* **Project Status:**
* We implemented Student Report card System with a minimal working
* You can do following actions,

1. Add Student Record
2. Display All Student Records
3. Deleting Student Record
4. Search Specific Record
5. Modify Student Record

* user based on their categories.
* **Existing system:-** Large amount of paper work is needed to accomplish the task in the current system and more time is needed to perpetuate
* necessary record. This system is used far and wide and it requires lot of human efforts and interaction is needed to maintain
* the records. It is very difficult to get the information about the old students who left the college more than ten years. There
* is a chance of data loss in case of any physical damage is encountered or in case of any natural disaster. It is decentralised
* and asymmetric way of storing the data. User has to wait in the queue to get his details and he has to visit the centre during
* the business hours. So it will take more time. In this case data manipulation, data hacking can be done easily. There is low
* level of data security and high level of data threatening. Parents’ will not able to get the status of the student until and
* unless they visit the college. There is no transparency regarding the student information.

**Proposed system:-**

**Current system has encountered with many limitations. The suggested system hits the restrictions found in the current**

**system. There exists lot of advantages in a proposed system that is the system is complete package so there is no slipups**

**concerning the statistics of student’s attendance and internal assessment marks etc. Also it eliminates redundancy. It is easy**

**to get information from single source. No latency when bringing back statistics and status of the student. Complete**

**statistics of a student can be recovered in the method of SMS to the mobile phone through one touch.**

**Notifications will be conveyed to the respective guardians through SMS when particular student not able to attend the class**

**as per the university guidelines and also email can be sent to respective parent and also pie and bar charts can be generated**

**as per student’s performance. Also it allows admin to store student and faculty details also admin can track the faculty and**

**student details. The application allows bringing about courses surrounded by a class domain. It has got profile based super**

**vision like administrator, faculty and student. It allows faculties to manage student attendance and their internal assessment**

**marks. And also it allows students to view their own details and also can generate report. Students get alerted during**

**examination via circular. And also it allows storing internal assessment marks and also generates the average.**

**ANALYSIS:-**

* **Introduction:-**
* **Advantages and disadvantages:-**
* **Requirements specifications:-**
  + - **Software requirements:-**
    - **Hardware requirements:**
* **Flowchart :-**
* **IMPLEMENTATION AND DESIGN:-**

**Introduction:-**

**Front-End:-** Naturally the first thing to do is create a simple code in C and C++. The features of this Application development will address the following user stories.

* As a user, I can use this application.
* As a user, I can analyse the Student information.
* As a user, I can keep track of how many students records.

As a user, I can see the credit scores or student

**Back-End:-**

**Testing and validation:-** Testing is the major control measure used during software development . It’s basic function is to detect errors in the software. During requirement analysis and design , the output is a document that is usually a textual and no executable. After the coding phase , computer programs are available that can be executed for testing purpose. This implies that testing not only, has to uncover errors introduced during coding, but also errors introduced during previous phase. Thus the goal of testing is to uncover the requirements, design and coding errors in the programs. So after testing the outputs of my project are as follows:

Implementation  
Once they receive the credentials they are able to access the application. And also to monitor the statistics like they can

generate their own report of internal assessments and attendance and they can view their performance. Figure 1 explains

about how data get distributed and processed in the application. It has got four modules. They are Login Module, Student

Data Module, SMS module, Email module, Report module. End user will be going to login to the system. There are three

types of users are created based on the access criteria. They are Admin, Faculty and Student/Guardian. Admin users are

responsible for storing student data and faculty details, scheduling time table for faculties. Admin users can create, replace,

update, delete (CRUD) the student and faculty details.

Admin users can also generate all types of reports which are available in the application. Admin user can send circular or

notice to faculties and students. Admin users will have all the privileges and access to the application. Admin user is a root

user; he is having complete control over the application. Faculty users are responsible for maintaining the attendance of the

students, internal assessments marks, generating time tables. Faculty users can generate report based on the student

progress like internal assessment, attendance. Student/Guardian users are responsible for viewing their own page; they can

check their academic progress, and attendance. They are able to generate report and charts like bar/pie based on their

performance. Once particular user is positively registered in, if logged in user is student then he is not authorise to make

changes or manipulate the details. If logged in user is a faculty user then he can manipulate student attendance and internal

assessment marks. If logged in user is an admin user then he is the only person to manipulate all the necessary details

regarding students and faculties. When an admin finishes all manipulations, faculty will generate the report and he will be

sending an email or SMS to the respective student’s parent.

**THE PROJECT SOURCE CODE :-**

**#include<fstream.h>**

**#include<iostream.h>**

**#include<conio.h>**

**#include<string.h>**

**#include<stdio.h>**

**#include<stdlib.h>**

**class stud**

**{**

**char name[100];**

**int rollno;**

**int emiii,ds,oopj,lst,dis;**

**int per;**

**void calculate()**

**{ per=(emiii+ds+oopj+lst+dis)\*0.2;}**

**char grade;**

**void calgrade()**

**{**

**if(per>=90 && per<=100)**

**{**

**grade='A';**

**}**

**if(per>=70 && per<90)**

**{**

**grade='B';**

**}**

**if(per>=50 && per<70)**

**{**

**grade='C';**

**}**

**if(per<50)**

**{**

**grade='D';**

**}**

**}**

**public:**

**void getdata();**

**void showdata();**

**int getrollno();**

**void modify();**

**};**

**void stud::modify()**

**{char chi;**

**cout<<" What do you want to modify ? "<<"\n";**

**cout<<" 1. Roll number: "<<"\n";**

**cout<<" 2. NAME: "<<"\n";**

**cout<<" 3. Marks in Eng..Mathematics III: "<<"\n";**

**cout<<" 4. Marks in Data Structures: "<<"\n";**

**cout<<" 5. Marks in OOPs Thru..Java "<<"\n";**

**cout<<" 6. Marks in Logic & Switching Theory: "<<"\n";**

**cout<<" 7. Marks in Discrete Structures: "<<"\n";**

**cout<<" Enter your choice: "<<"\n";**

**cin>>chi;**

**switch(chi)**

**{**

**case '1':**

**cout<<" The existing roll no. is "<<rollno<<"\n";**

**cout<<" Enter your new roll no ";**

**cin>>rollno;**

**cout<<"\n"<<" Your data has been modified ";**

**break;**

**case '2':**

**cout<<" The existing name is "<<name<<"\n";**

**cout<<" Enter your new name ";**

**gets(name);**

**cout<<"\n"<<" Your data has been modified ";**

**break;**

**case '3':**

**cout<<" The existing marks in Eng..Mathematics III is: "<<emiii<<"\n";**

**cout<<" Enter your new marks in Eng..Mathematics III: ";**

**cin>>emiii;**

**cout<<"\n"<<" Your data has been modified ";**

**break;**

**case '4':**

**cout<<" The existing marks in Data Structures is: "<<ds<<"\n";**

**cout<<" Enter your new marks in Data Structures: ";**

**cin>>ds;**

**cout<<"\n"<<" Your data has been modified ";**

**break;**

**case '5':**

**cout<<" The existing marks in OOPs Thru..Java is: "<<oopj<<"\n";**

**cout<<" Enter your new marks in OOPs Thru..Java ";**

**cin>>oopj;**

**cout<<"\n"<<" Your data has been modified ";**

**break;**

**case '6':**

**cout<<" The existing marks in Logic & Switching Theory is: "<<lst<<"\n";**

**cout<<" Enter your new marks in Logic & Switching Theory: ";**

**cin>>lst;**

**cout<<"\n"<<" Your data has been modified ";**

**break;**

**case '7':**

**cout<<" The existing marks in Discrete Structures is: "<<dis<<"\n";**

**cout<<" Enter your new marks in Discrete Structures: ";**

**cin>>dis;**

**cout<<"\n"<<" Your data has been modified ";**

**break;**

**default :**

**cout<<"Wrong Choice::Invalid\_Input ";**

**break;**

**}**

**calculate();**

**calgrade();**

**}**

**void stud::getdata()**

**{**

**cout<<"\n\n\n\n\t\tStudent record.\n\n";**

**cout<<"\tEnter your name:";**

**gets(name);**

**cout<<"\n\tEnter rollno ";**

**cin>>rollno;**

**cout<<"\n\tEnter marks in Eng..Mathematics III: ";**

**cin>>emiii;**

**cout<<"\n\tEnter marks in Data Structures: ";**

**cin>>ds;**

**cout<<"\n\tEnter marks in Oops Through..Java: ";**

**cin>>oopj;**

**cout<<"\n\tEnter marks in Logic & Switching Theory: ";**

**cin>>lst;**

**cout<<"\n\tEnter marks in Discrete Structures:" ;**

**cin>>dis;**

**cout<<"\n";**

**calculate();**

**calgrade();**

**}**

**void stud::showdata()**

**{**

**cout<<"\n\tName is "<<name;**

**cout<<"\n\tRoll no= "<<rollno;**

**cout<<"\n\tMarks in 5 subjects are: "<<emiii<<"\t"<<ds<<"\t"<<oopj<<"\t"<<lst<<"\t"<<dis;**

**cout<<"\n\tpercentage is "<<per;**

**cout<<"\n\tGrade is "<<grade;**

**//cout<<"==============================================================";**

**}**

**int stud::getrollno()**

**{**

**return rollno;**

**}**

**//prototype definition**

**void gdata();**

**void sdata();**

**void sedata();**

**void ddata();**

**void intro();**

**void mdata();**

**// void sodata();**

**class login**

**{ public:**

**char name[100];**

**char uname[100];**

**char pass[100];**

**void getdataa();**

**void searchdataa();**

**};**

**void introa();**

**void gdataa();**

**void sedataa();**

**void login::getdataa()**

**{**

**cout<<"\n\n\n\n\t\tSIGN UP.\n\n";**

**cout<<"\tEnter your name: ";**

**gets(uname);**

**cout<<"\n\tEnter your id: ";**

**gets(name);**

**cout<<"\n\tEnter password: ";**

**gets(pass);**

**}**

**void gdataa()**

**{login s1;**

**s1.getdataa();**

**ofstream outfile;**

**outfile.open("l1.dat",ios::app|ios::binary);**

**outfile.write((char \*)&s1,sizeof(login));**

**outfile.close();**

**cout<<"\n\n\t\tSigned up successfully";**

**getche();**

**}**

**void sedataa()**

**{**

**login s1;**

**ifstream fin;**

**char un[100],rn[100],ps[100],found='n';**

**fin.open("l1.dat",ios::binary);**

**cout<<"\n\n\tEnter user id to be login\n\t\t::";**

**gets(rn);**

**while(fin.read((char\*)&s1,sizeof(login)))**

**{**

**if(strcmp(rn,s1.name)==0)**

**{ lb:**

**strcpy(un,s1.uname);**

**cout<<"\n\n\tEnter password::";**

**gets(ps);**

**if(strcmp(ps,s1.pass)==0)**

**{**

**clrscr();**

**cout<<"\n\tWELCOME\n";**

**cout<<"\n\tUSERNAME: ";**

**puts(un); char ch;**

**intro();**

**do //DO STARTS**

**{**

**clrscr();**

**cout<<"\n\n\tMAIN MENU";**

**cout<<"\n\n\t01.Create Student Record.";**

**cout<<"\n\n\t02.DISPLAY Student Record.";**

**cout<<"\n\n\t03.SEARCH Student Record.";**

**cout<<"\n\n\t04.Delete Student Record.";**

**cout<<"\n\n\t05.Modify Student Record.";**

**cout<<"\n\n\t06.EXIT.";**

**cout<<"\n\n\t\tEnter your choice::";**

**cin>>ch;**

**clrscr();**

**switch(ch) //SWITCH STARTS**

**{**

**case '1': gdata();**

**break;**

**case '2': sdata();**

**break;**

**case '3': sedata();**

**break;**

**case '4': ddata();**

**break;**

**case '5': mdata();**

**break;**

**case '6' : exit(6);**

**break;**

**default: cout<<"\a";**

**break;**

**} //SWITCH COLSED**

**}while(ch>='1' && ch<='6');**

**/////////////////////////////////////////////////////////////////////////**

**}**

**else**

**{**

**cout<<"\n\n\n\n\n\t\t\t\tWrong password..........";**

**goto lb;**

**}**

**found='y';**

**break;**

**}**

**}**

**if(found=='n')**

**cout<<"\nNEW USER!!!!!!!\n";**

**fin.close();**

**getche();**

**}**

**void introa()**

**{**

**cout<<"\n\n\n\t\t STUDENT REPORT CARD SYSTEM";**

**cout<<"\n\n\n\tDONE BY : BHARATH DURGAM (160117733320), SHIVA PRASAD REDDY (160117733113)";**

**cout<<"\n\tCollege : Chaitanya Bharathi Institute of Technology";**

**cout<<"\n\n\tSIGN IN for main menu....";**

**getch();**

**}**

**void gdata()**

**{stud s1;**

**s1.getdata();**

**ofstream outfile;**

**outfile.open("stud1.dat",ios::app|ios::binary);**

**outfile.write((char \*)&s1,sizeof(stud));**

**outfile.close();**

**cout<<"\n\n\t\tRecord inserted successfully";**

**getche();**

**}**

**void sdata()**

**{stud s1;**

**ifstream fin;**

**fin.open("stud1.dat",ios::binary);**

**while(fin.read((char\*)&s1,sizeof(stud)))**

**{**

**s1.showdata();**

**cout<<"\n\t===========================================================\n";**

**}**

**fin.close();**

**getche();**

**}**

**void sedata()**

**{**

**stud s1;**

**ifstream fin;**

**int rn,found='n';**

**fin.open("stud1.dat",ios::binary);**

**cout<<"\n\n\tEnter roll no. to be searched\n\t\t\t";**

**cin>>rn;**

**while(fin.read((char\*)&s1,sizeof(stud)))**

**{**

**if(s1.getrollno()==rn)**

**{**

**s1.showdata();**

**found='y';**

**break;**

**}**

**}**

**if(found=='n')**

**cout<<"\nrecord donot exists\n";**

**fin.close();**

**getche();**

**}**

**void ddata()**

**{**

**int n;**

**cout<<"\n\n\tEnter roll no to be deleted......";**

**cin>>n;**

**ifstream ifile;**

**ifile.open("stud1.dat",ios::binary);**

**ofstream ofile;**

**ofile.open("temp2.dat",ios::app|ios::binary);**

**stud s1;**

**while(ifile.read((char\*)&s1,sizeof(s1)))**

**{**

**if(s1.getrollno()!=n)**

**{**

**ofile.write((char\*)&s1,sizeof(s1));**

**}**

**else**

**{**

**cout<<"The record has been deleted ";**

**getche();**

**}**

**}**

**ifile.close();**

**ofile.close();**

**remove("stud1.dat") ;**

**rename("temp2.dat","stud1.dat");**

**}**

**void intro()**

**{**

**cout<<"\n\n\n\t\t STUDENT";**

**cout<<"\n\n\t\tREPORT CARD";**

**cout<<"\n\n\tPress enter for main menu....";**

**getch();**

**}**

**void mdata()**

**{int n;**

**cout<<"\n\n\tEnter roll no to be modified......";**

**cin>>n;**

**ifstream ifile;**

**ifile.open("stud1.dat",ios::binary);**

**ofstream ofile;**

**ofile.open("temp2.dat",ios::app|ios::binary);**

**stud s1;**

**while(ifile.read((char\*)&s1,sizeof(s1)))**

**{**

**if(s1.getrollno()==n)**

**{ s1.modify();**

**ofile.write((char\*)&s1,sizeof(s1));**

**}**

**else**

**{**

**ofile.write((char\*)&s1,sizeof(s1));**

**}**

**}**

**ifile.close();**

**ofile.close();**

**remove("stud1.dat") ;**

**rename("temp2.dat","stud1.dat");**

**}**

**///////////////////////////////sorting///////////////**

**/\*void sodata()**

**{**

**int a;**

**stud s1;**

**ifstream ifile;**

**ofstream ofile;**

**ifile.open("stud1.dat",ios::binary);**

**ofile.open("temp2.dat",ios::binary|ios::app);**

**ifile.seekg(0|ios::beg);**

**ifile.read((char\*)&s1,sizeof(s1));**

**a=s1.getrollno();**

**while(ifile.read((char\*)&s1,sizeof(s1)))**

**{**

**if(a<s1.getrollno())**

**{**

**a=s1.getrollno();**

**}**

**}**

**cout<<"\n\n\n\n\t\t";**

**cout<<a<<"\n";**

**ifile.close();**

**for(int i=1;i<=a;i++)**

**{**

**ifile.open("stud1.dat",ios::binary);**

**ifile.seekg(0|ios::beg);**

**while(ifile.read((char\*)&s1,sizeof(s1)))**

**{**

**if(s1.getrollno()==i)**

**{**

**ofile.write((char \*)&s1,sizeof(s1));**

**}**

**}**

**ifile.close();**

**}**

**ofile.close();**

**cout<<"\t\t Data sorted ";**

**remove("stud1.dat");**

**rename("temp2.dat","stud1.dat");**

**getche();**

**}**

**\*/**

**void main()**

**{**

**char cha;**

**introa();**

**do //DO STARTS**

**{**

**clrscr();**

**cout<<"\n\n\tUSER MENU";**

**cout<<"\n\n\t01.SIGN UP";**

**cout<<"\n\n\t02.SIGN IN";**

**cout<<"\n\n\t03.EXIT.";**

**cout<<"\n\n\t\tEnter your choice::";**

**cin>>cha;**

**clrscr();**

**switch(cha) //SWITCH STARTS**

**{**

**case '1': gdataa();**

**break;**

**case '2': sedataa();**

**break;**

**case '3':**

**exit(3);**

**break;**

**default: cout<<"\a";**

**break;**

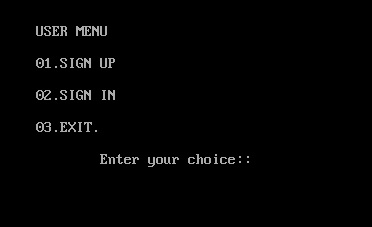
**} //SWITCH COLSED**

**}while(cha>='1' && cha<='3');**

**}**

**SAMPLE OUTPUTS :-**

**USER MENU :-**



**SIGN UP :-**



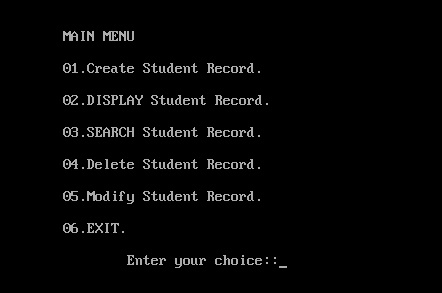
**LOGIN PAGE :-**



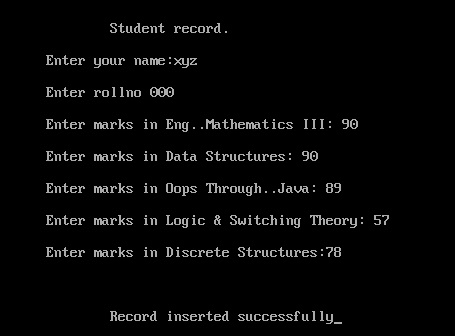
**WELCOME PAGE :-**



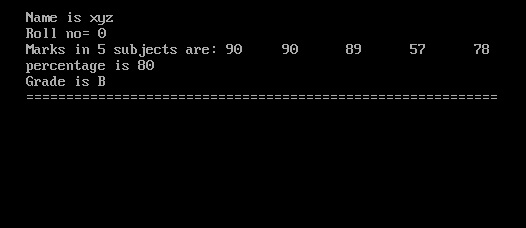
**MAIN MENU :-**



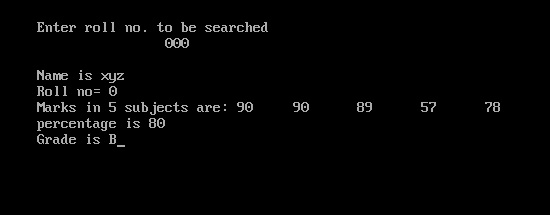
**CREATING STUDENT RECORD :-**



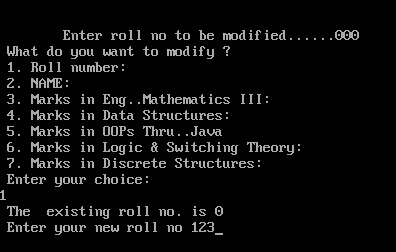
**DISPLAY ALL RECORDS :-**



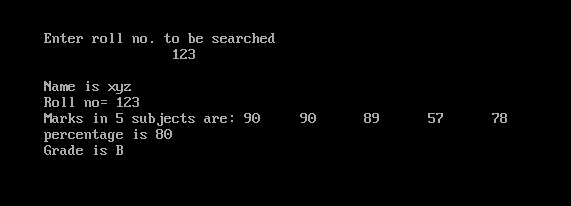
**SEARCHING SPECIFIC RECORD :-**



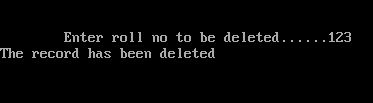
**MODIFY STUDENT RECORD :-**



**AFTER ALTERED :-**



**DELELTE RECORD :-**



**CONCLUSION :-**

**Student Management System is very useful in an institution or in college or in universities. There is no paper work in this**

**proposed system. Supervision can be done from anywhere. This project especially minimizes human effort necessary. This**

**application is handled by the college so there is no information leak and data will be secured. Since it is a web based**

**application anyone can use the system anywhere at any time and it is very easy to get the necessary information without the**

**latency. It is very useful to the students to get their report on attendance and internal assessments. Parents also get benefited**

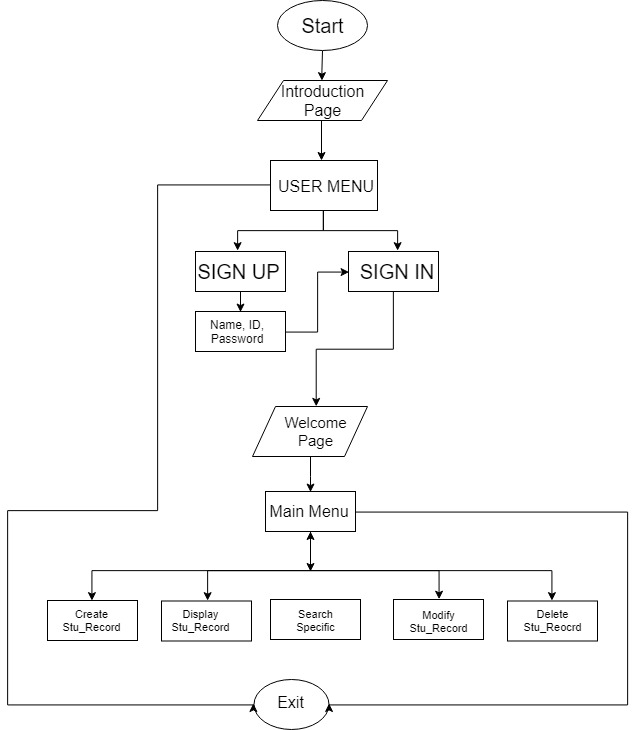
**more since college is going to send the notification of the student via the SMS or email will be sent to get the recent**

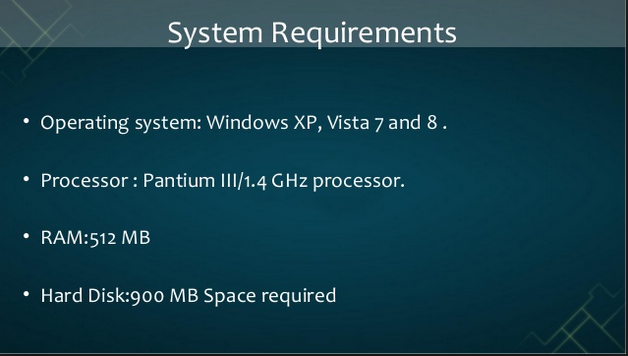
**activities happen in the college. Since this application will be handled by the college whenever they need any changes in an**

**application they can make it without the upfront investment, and the system will be more secure when it is handled by the**

**own college.**

**FLOWCHART :-**





PRESENTED BY..

MOHAMMAD ABDUL IMRAN (17R01A05E7)

MOHD SUBHAN (17R01A05E9)

MANDALA SHIVA KUMAR (18R05A0525)

SAARA SANTOSH GOUD (18R05A0535)