**Abstract**

Our project is about student exam management system. The main objective of the system is to gather student data, display the student details, checking the eligibility status of the student (fee payment, attendance), accepting the marks and finally displaying the marks.

* Segregating and checking the student’s data for eligibility to give final exams.
* The criteria for eligibility to give final exams are minimum of 65% attendance and fee payment status.
* Display of results status, that will be pass or fail after giving the input of marks of 5 subjects in previous exams.
* Display of fee payment status that is ‘P’ or ‘N’, ‘P’ for paid status and ‘N’ for rest of the cases.
* Display of student’s details as in hall ticket number, student name, student address, student class.
* Display of percentage of attendance after giving the number of days present. As, total number of working days will be predefined.
* Taking the input as strength of the class to design the code to read the data of the entire class.

**Declaration**

We, **Abhiram k (2010080001), Sahasi korraparti(2010080017) and Kapila**

**Vishnu vamsi(2010080036)** Students of I B.Tech-Sem-I(regular batch) of KL

University hereby declare that the project work presented in this report is us

own work.

This work has not been previously submitted to any other university for any

examinations.

Date:

Abhiram K (2010080001)

Place: Hyderabad Sahasi Korrapati (2010080017)

Kapila Vishnu Vamsi(2010080036)

**Acknowledgement**

We are greatly indebted to the authorities of KL deemed to be university Hyderabad . For providing us the necessary facilities to successfully completion of this project.

We thank and express our solicit gratitude to  **Mrs. Girija Rani Suthoju course coordinator,** for her valuable help and support which helped us a lot in successfully completing our mini project.

We would like to express our sincere gratitude to our principal **Prof.L.Koteswar rao** for providing the necessary infrastructure to complete our project. Finally, we would like to express our heartfelt thanks to our parents who were very supportive both financially and mentally and for their encouragement to achieve our goals.

**Index**

1. Certificate ii
2. Project abstract 1
3. Declaration 2
4. Acknowledgement 3
5. Index 4
6. Introduction 5-10
7. Description 8-19

Algorithm 8-10

Flowchart 11-17

Validation 18-19

1. Source code 20-28
2. Project output screenshots 29-31
3. Conclusion 32
4. Bibliography 33

**Introduction**

Our project is about student exam management system. The main objective of the system is to check the eligibility status of students to allow them to write the final exams.

Our Goal is to modify the existing manual storage of the records of students which will reduce lot of human energy and using computer knowledge to make things smoother and faster simultaneously.

**Existing System**

The existing system is the old hand, manual written which will have much error

Disadvantages:-

* It contains much error because of manual entries.
* It takes much time to store and collect the data as we need to write manually every record.
* Even though we write the data it is difficult to preserve the data as it needs many shelves to keep the files and fire accidents can destroy every record as we saw many cases like that which destroyed many records.
* Records can be manipulated if it is written manually, etc.

**Proposed System**

The proposed system is written in C language which is easy and efficient to understand by the user. It is much error free than the manual hand written record.

Advantages: -

* It is easy to write the code.
* It is error free.
* It takes less time and is more efficient to use the proposed system.
* It can store more records than the existing one within less time.
* It is easy to store the data in this system.
* Accuracy to work and easy to update information

**Hardware and Software Requirement Specifications**

1. Hardware Requirement

* Processor : Windows 10
* RAM: 6gb Ram or Above
* Hard Disk : 256GB HDD or Above

1. Software Requirement

* Tools : DEV-C++
* Language: C language

In our code we tried to create a software which takes the inputs as student’s details, paid status, attendance status and checks the student is eligible or not on the criteria we specified.

The criteria we specified are that a student is only eligible to give the final exams if and only if his fee payment status is ‘P’ and the percentage of attendance is more than 65%.

We get the status of fee payment when the fees is fully paid ‘P’ and for rest of the cases it is ‘F’. similarly we get the status of result – ‘PASS’ if the student gets greater than or equal to 50% in all the subjects , else the student gets ‘FAIL’ as result. The status of attendance is also achieved by giving number of days present by the student. if the percentage is greater than or equal to 65%, the student is allowed to take the exam.

In the source code the total work is divided into 6 functions.

The functions are:

1. Get student data.
2. Display student data.
3. Check the eligibility of student data.
4. Accept the marks.
5. Display the result.
6. Exit.

The functionalities of defined functions:

1. Get student data :

Firstly, here we have to specify the strength of the class. So, that the software reads the data of the entire class accordingly. We need to give hall ticket number, student name, student address, student class, student fees status ‘P’ for paid and ‘N’ for rest of the cases and the number of days present for each student in the class.

1. Display student data :

In the display student data function, student hall ticket number, student name, student address, student class, student fees status ‘P’ for paid and ‘N’ for rest of the cases and the number of days present are displayed.

1. Check eligibility :

In this function the student’s details hall ticket number, student name, class, attendance percentage, fees status, eligibility are displayed. Percentage for attendance is given as (student’s number of days present/number of working days)\*100. If percentage is greater than or equal to 65 and students fee status is P. the student will be eligible to proceed further with the process else the student will be declared not eligible.

1. Accept marks :

The marks are only taken for the students who are declared to be eligible. The marks are to be given for five subjects out of 100 for each.

1. Display the result :

Result is displayed for the students who were declared as eligible only. The students gets ‘PASS’ if he gets greater than or equal to 50 marks in all subjects else the student gets ‘FAIL’.

1. Exit :

Entering 6 will exit the code.

We used c language to do this project since it contains all the commands required for execution of program. In c language we used the concepts of loops, strings, structures and functions. The libraries we used are standard library, conio.h, string.h, stdlib.h.

**Algorithm**

**Main Function Algorithm**

**Step 1:** Start

**Step 2**: Declare required variables

**Step 3**: Input Student Data

**Step 4**: If attendance is greater or equals to 65, then check fee status, otherwise END

**Step 5**: If the fee is paid then the student is eligible for exam otherwise not eligible

**Step 6**: If eligible, Then Enter subject marks

**Step 7**: Display the result

**Step 8**: If marks are greater than or equals to 50% in every subject then display “PASS” otherwise display “FAIL”

**Step 9**: Stop

1. **Get Student Data Function Algorithm**

**Step 1:** Start

**Step 2**: Accept student Hall Ticket Number

**Step 3**: Accept student name

**Step 4**: Accept student address

**Step 5**: Accept Fee status

**Step 6**: Accept number of days present

**Step 7**: Stop

1. **Display Student Data Function Algorithm**

Step 1: Start

Step 2: Displaying the Student data

* Display Hall Ticket number
* Displaying Student name
* Displaying Fee status
* Displaying Student attendance

Step 3: Stop

1. **Checking Eligibility Function Algorithm**

Step 1: Start

Step 2: Calculating the attendance percentage

Step 3: If the attendance percentage is greater than or equals to 65 and the fee is paid then display “Eligible and Paid”

Step 4: If the attendance percentage is less than 65 and the fee is paid then display “Not eligible and Paid”

Step 5: If the attendance percentage is less than 65 and the fee is not paid then display “Not eligible and Not Paid”

1. **Get Student Result Function Algorithm**

Step 1: Start

Step 2: Accept Subject 1 marks

Step 3: Accept Subject 2 marks

Step 4: Accept Subject 3 marks

Step 5: Accept Subject 4 marks

Step 6: Accept Subject 5 marks

Step 7: Stop

1. **Calculate Student Result Function Algorithm**

Step 1: Start

Step 2: If the subject 1 marks are greater than or equals to 50% then check for the remaining subject marks otherwise display fail

Step 3: If the subject 2 marks are greater than or equals to 50% then check for the remaining subject marks otherwise display fail

Step 4: If the subject 3 marks are greater than or equals to 50% then check for the remaining subject marks otherwise display fail

Step 5: If the subject 4 marks are greater than or equals to 50% then check for the remaining subject marks otherwise display fail

Step 6: If the subject 5 marks are greater than or equals to 50% then “PASS” otherwise display “FAIL”

Step 7: Stop

1. **Display Student Result Function Algorithm**

Step 1: Start

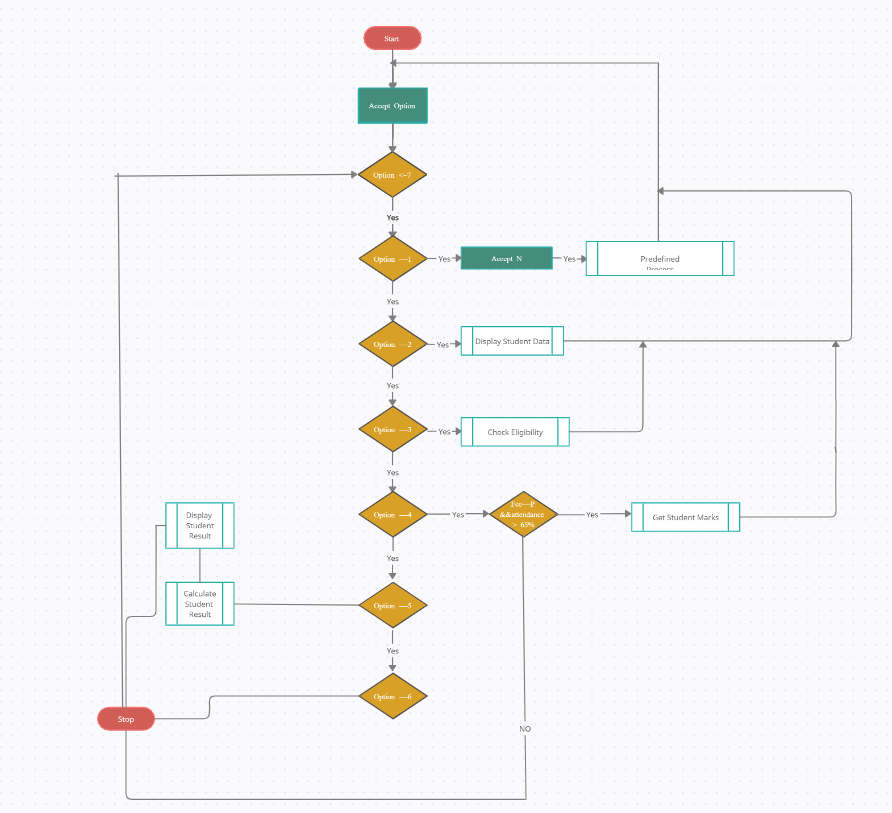
Step 2: Displaying Student results

* Displaying Student Hall Ticket Number
* Displaying Student Name
* Displaying Student Result

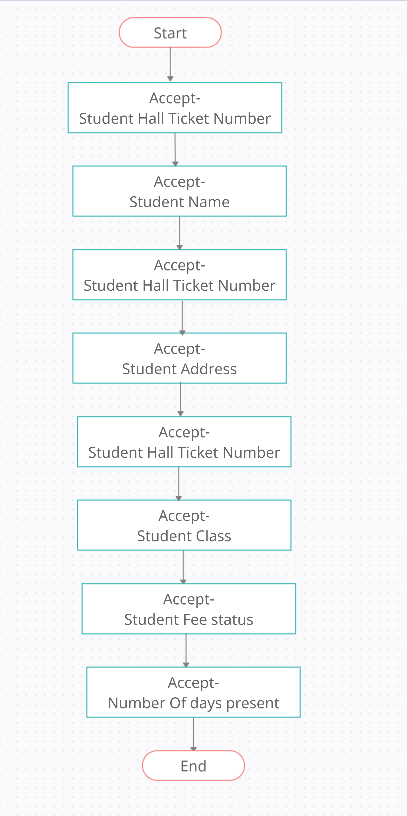
Step 3: Stop

**Flowchart**

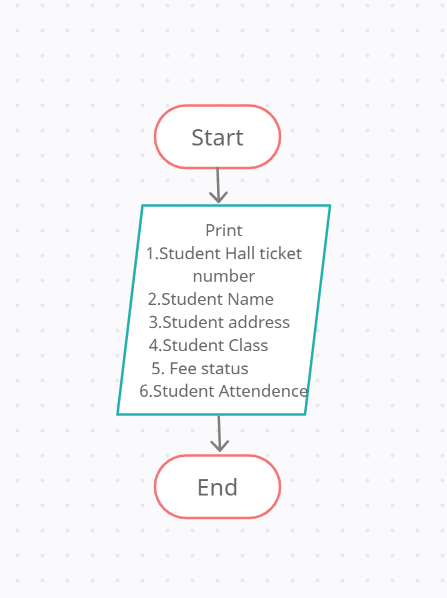
1. **Main Function Flow Chart**

****

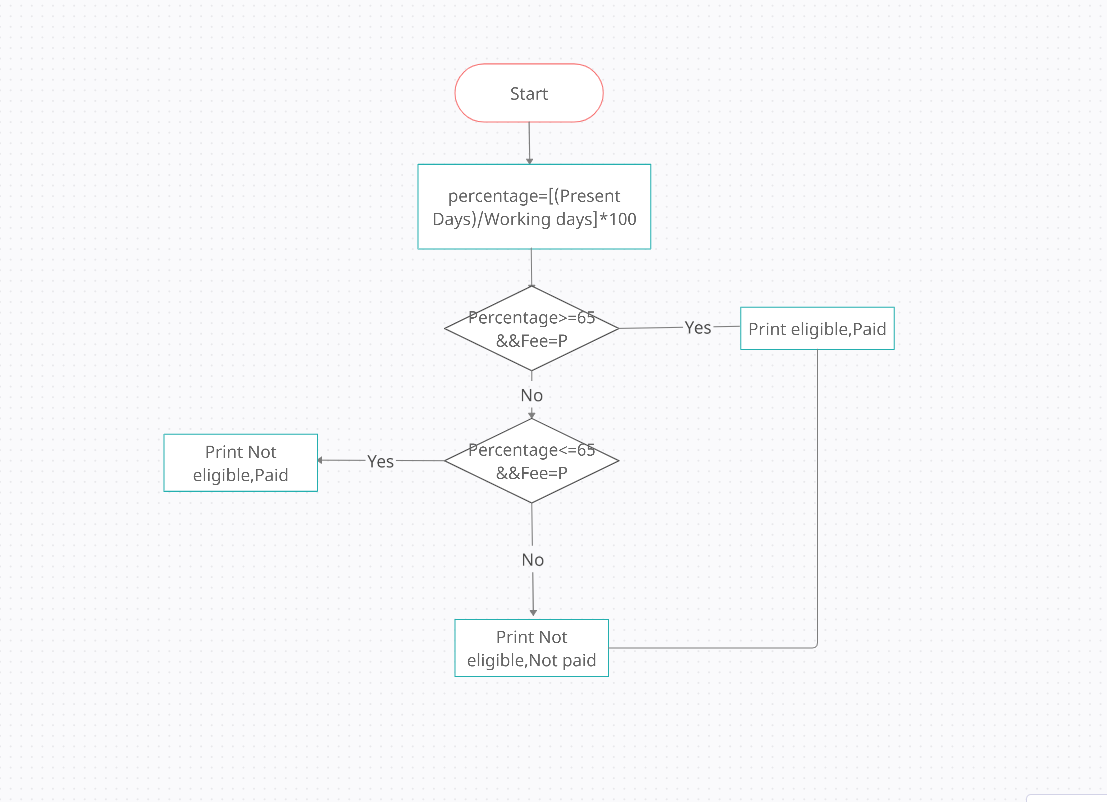
1. **Get Student Data Function Flow Chart**

****

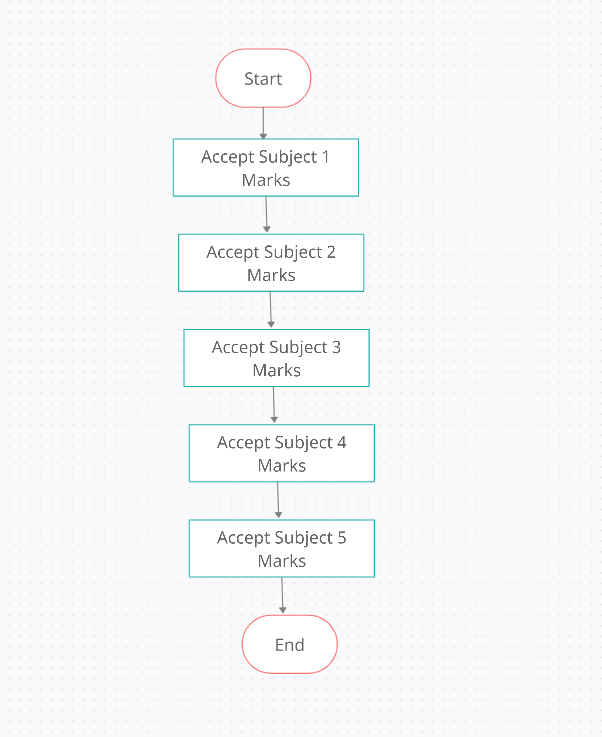
1. **Display Student Data Flow Chart**

****

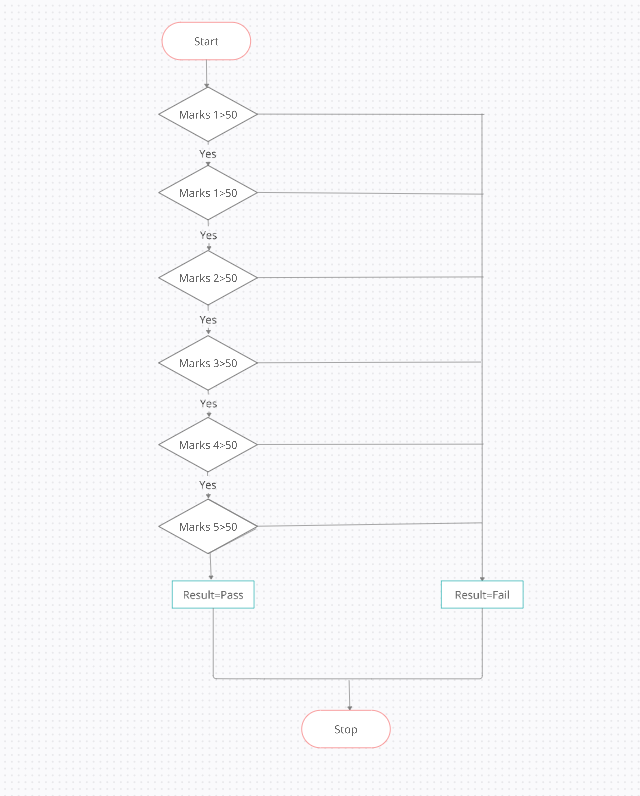
1. **Check Eligibility Function Flow Chart**

****

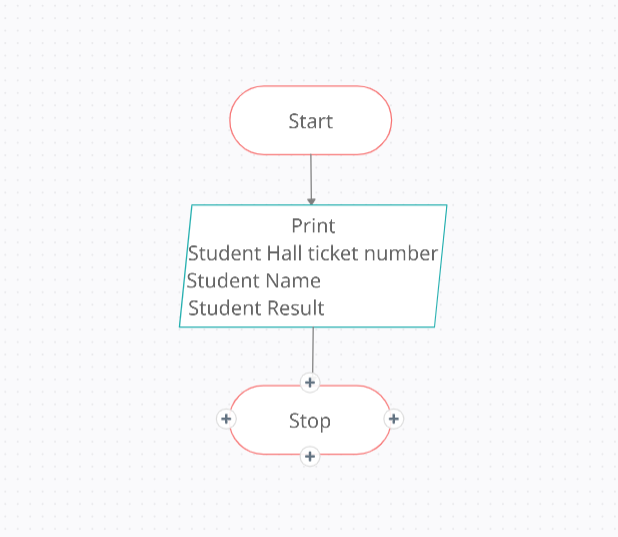
1. **Get Student Result Function Flow Chart**

****

1. **Calculate Student Result Flow Chart**

****

1. **Display Student Result Flow Chart**

****

**Project Input and Output validation**

**Choice**

The choice entered must be between 1 and 6 in the menu of functions

**Get student data**

The strength of the class must be less than or equal to 100.

The maximum characters in hall ticket number of student can be 10.

The maximum characters in name of the of student can be 50.

The maximum characters in address of student can be 50.

The maximum characters in class of student can be 25.

The fee payment status must be given uppercase P or N.

Data of all the student must be entered as one student after another.

No spaces between words must be given while entering the data a – should be used.

**Display student data**

For this function to work step Get student data function must be executed.

**Check eligibility**

For this function to work step Get student data function must be executed.

**Get student result**

For this function to work step Get student data function must be executed.

The results should be entered in the order of eligible students, which can be obtained from check eligibility function.

The marks should be less than or equal to 100.

If the student was absent for exam enter marks as zero.

**Display student result**

For this function to work steps Get student data & Getstudentresultfunction must be executed.

**Source Code**

**#include<stdio.h>**

**#include<conio.h>**

**#include<string.h>**

**#include<stdlib.h>**

**#define NO\_OF\_WORKING\_DAYS 100**

**#define MAX\_SIZE 100**

struct StudentResult

{

int marks1;

int marks2;

int marks3;

int marks4;

int marks5;

char result[10];

};

struct StudentMaster

{

char stud\_hallticket[10];

char stud\_name[50];

char stud\_address[50];

char stud\_class[25];

char stud\_fees\_status[1];

int number\_of\_days\_present;

struct StudentResult sm;

};

struct StudentMaster stud[MAX\_SIZE];

void getStudentResult(struct StudentResult \*s)

{

printf("\t ------------------------------------\n") ;

printf("\t Enter the Subject 1 Marks\n");

scanf("%d",&s->marks1);

printf("\t Enter the Subject 2 Marks\n");

scanf("%d",&s->marks2);

printf("\t Enter the Subject 3 Marks\n");

scanf("%d",&s->marks3);

printf("\t Enter the Subject 4 Marks\n");

scanf("%d",&s->marks4);

printf("\t Enter the Subject 5 Marks\n");

scanf("%d",&s->marks5);

}

void calculateStudentResult(struct StudentResult \*s)

{

if(s->marks1 >=50 && s->marks2 >=50 && s->marks3 >=50 &&

s->marks4 >=50 && s->marks5 >=50)

{

strcpy(s->result,"PASS");

}

else

{

strcpy(s->result,"FAIL");

}

}

void displayStudentResult(struct StudentMaster obj)

{

printf("%s\t",obj.stud\_hallticket);

printf("%s\t",obj.stud\_name);

printf("%s\n",obj.sm.result);

}

void getStudentData(struct StudentMaster \*obj)

{

printf("\t ------------------------------------\n") ;

printf("\t Enter the Student Hall Ticket Number\n");

scanf("%s",obj->stud\_hallticket);

printf("\t Enter the Student Name\n");

scanf("%s",obj->stud\_name);

printf("\t Enter the Student Address\n");

scanf("%s",obj->stud\_address);

printf("\t Enter the Student Class\n");

scanf("%s",obj->stud\_class);

printf("\t Enter the Student Fees Status P for Paid and N for Not Paid\n");

scanf("%s",obj->stud\_fees\_status);

printf("\t Enter the Student Number of days Present\n");

scanf("%d",&obj->number\_of\_days\_present);

}

void displayData(struct StudentMaster obj1)

{

printf("\t ------------------------------------\n") ;

printf("The Student Hall Ticket Number:\t");

printf("%s\n",obj1.stud\_hallticket);

printf(" The Student Name:\t");

printf("%s \n",obj1.stud\_name);

printf(" The Student Address:\t");

printf("%s \n",obj1.stud\_address);

printf("The Student Class:\t");

printf("%s \n",obj1.stud\_class);

printf("The Student Fees Status P for Paid and N for Not Paid:\t");

printf("%c \n",obj1.stud\_fees\_status[0]);

printf(" The Student Number of days Present:\t");

printf("%d \n",obj1.number\_of\_days\_present);

}

void checkElgibility(int n1)

{

int i;

float percentage;

printf("Hall Ticket Number\tStudent Name\tClass\tAttendence Percentage\tFeesStatus\tEligibility\n");

printf("---------------------------------------------------------------------------------------\n");

for(i=0;i<=(n1-1);i++)

{

percentage=(stud[i].number\_of\_days\_present/(float)NO\_OF\_WORKING\_DAYS)\*100;

if(percentage>=65 && stud[i].stud\_fees\_status[0]=='P')

{

printf("%s \t \t \t%s \t %s \t %2f \t PAID \t Eligible\n",stud[i].stud\_hallticket,stud[i].stud\_name,stud[i].stud\_class,percentage);

}

else if(percentage<=65 && stud[i].stud\_fees\_status[0]=='P')

{

printf("%s \t \t \t%s\t %s \t %2f \t PAID \t Not Eligible\n",stud[i].stud\_hallticket,stud[i].stud\_name,stud[i].stud\_class,percentage); }

else

{

printf("%s \t \t\t %s\t %s \t %2f \t NOT PAID \t Not Eligible\n",stud[i].stud\_hallticket,stud[i].stud\_name,stud[i].stud\_class,percentage);

}

printf("------------------------------------------------------------------------------------------\n");

}

}

int main()

{

int n,i,option;

float percentage;

do

{

printf("STUDENT EXAMINATION SYSTEM\n");

printf("---------------------------\n");

printf("Enter 1 to Get Student Data\n");

printf("Enter 2 to Display Student Data\n");

printf("Enter 3 to Check Eligibility of Student\n");

printf("Enter 4 to Accept the Marks\n");

printf("Enter 5 to Display the Result\n");

printf("Enter 6 to Exit\n");

printf("Enter your choice\n");

scanf("%d",&option);

switch(option)

{

case 1:

printf("Enter the strength of the class less than or equal to 100\n");

scanf("%d",&n);

for(i=0;i<=(n-1);i++)

{

getStudentData(&stud[i]);

}

break;

case 2:

for(i=0;i<=(n-1);i++)

{

displayData(stud[i]);

}

break;

case 3:

checkElgibility(n);

break;

case 4:

for(i=0;i<=(n-1);i++)

{ percentage=(stud[i].number\_of\_days\_present/(float)NO\_OF\_WORKING\_DAYS)\*100;

if(percentage>=65 && stud[i].stud\_fees\_status[0]=='P')

{

printf("Enter the marks of Student %s\n",stud[i].stud\_hallticket);

getStudentResult(&stud[i].sm);

}

}

break;

case 5:

for(i=0;i<=(n-1);i++)

{ percentage=(stud[i].number\_of\_days\_present/(float)NO\_OF\_WORKING\_DAYS)\*100;

if(percentage>=65 && stud[i].stud\_fees\_status[0]=='P')

{

calculateStudentResult(&stud[i].sm);

}

}

printf("\t Student Hall Ticket Number \t Student Name \t Student Result\n");

printf("\t ------------------------------------------------------------\n") ;

for(i=0;i<=(n-1);i++)

{ percentage=(stud[i].number\_of\_days\_present/(float)NO\_OF\_WORKING\_DAYS)\*100;

if(percentage>=65 && stud[i].stud\_fees\_status[0]=='P')

{

displayStudentResult(stud[i]);

}

}

break;

case 6:

exit(0);

break;

}

}while(option<=6);

return 0;

}

**Project output screenshots**

Menu screenshot

Text

Description automatically generated

Get student data screenshot

Text

Description automatically generated

Display student data screenshot

Text

Description automatically generated

Check eligibility screenshot

Text

Description automatically generated with low confidence

Get student marks screenshot

Text

Description automatically generated

Display result screenshot

Text

Description automatically generated with medium confidence

**Conclusion**

Hence a system will be developed which will help to manage the procedures single handedly which reduces errors and increase efficiency.

**Future Implementations**

In future implementations of this Student Exam Management System project can store data in database (Rows and Columns) by using SQL (Structured Query Language) and using commands like UPDATE, SELECT, DELETE, DROP, INSERT, etc, and by using high level languages like Java and Python which we can get the data of a particular student by just entering student unique ID

Hence a system will be developed which will help to manage the procedures single handedly.

**Bibliography**

* Yashwanth Kanitkar , “Let us c”,11th edition 2010.
* P Padmanabham , “C programming and data structures”,2nd edition 2005.
* <https://www.studentprojectguide.com/>
* <https://www.geeksforgeeks.org/>
* <https://www.quora.com/>