SCHOOL EXAMINATION SYSTEM

SUBMITTED BY

Tanishque kumar

Akshith gupta.

Ajay sai .k

OVERVIEW

The Examination Management System is a system which will be used as a platform for maintaining examination records of students. In this system you can enter the name, roll call, records and all other key identifiers to identify a particular pupil’s record

AIM

While the main objective of this project is to computerize the paperwork in the system and automate the work. The computerization is done so that the storage of all the details regarding students will be stored in the system which makes system centralized and the chance of duplication of any data is minimized. By doing this all the records of their grades of all the students will be available to every teacher. While by doing automation to the system will reduce the time for storing any data in the system

C program for the Examination Management System

```
include<stdio.h>
#include<stdlib.h>
#include<string.h>
int option=0;
int i=0;
int n=0;
int j=0;
float present=60.00;
char money='P';
float tdays=1;
struct student//Here
we are giving the
structure of the
Student.//
{
char name[20];
int Rollno;
char Fees;
float days;
float attendance;
}s[60];
```

```
//Here we are giving the
functions.//
void add(struct student s[]);
void eligible(struct student s[]);
void execute();
void printStudents(struct
student s[]);
void DeleteRecord(struct
student s[]);
//Now we are going to give the
function to execute the software
for the student exam
registeration.//
```

```
void execute()
{
printf("Enter the serial number
for selecting the option.\n");
printf("1) Eligible candidates\n ");
printf("2) To delete student
record\n ");
printf("3) To modifiy eligibility
criteria\n ");
printf("4) To reset the eligibility
criteria\n ");
printf("5) To show the list of all
students\n ");
printf("Enter 0 to exit\n ");
scanf("%d",&option);
```

```
// Switch Statement for
choosing the desired
option for the user //
switch (option)
{
case 1:
eligibleStudents(
s); execute();
break;
case 2:
deleteRecord(
s); execute();
break;
case 3:
printf("Old Attendance
required = %f", present);
printf("\n Enter the
updated attendence
required \n");
scanf("%f", &present);
printf("fees status required
was %c \n", money);
printf("Enter the new fees
status 'P' for paid 'N' for
not paid and 'B' for both
\n");
scanf("%c", &money);
printf("Eligibility Criteria
updated \n");
execute();
break;
```

```
case 4:
present = 75.00;
money = 'P';
printf("Eligibility creitria reset
\n");
execute();
break;
case 5:
printStudents(s);
execute();
break;
case 6:
execute();
break;
case 0:
exit(0);
default:
printf("Enter number only
from 0-4 \n");
execute();
```

```
/ Function to print the students record//
void printStudents(struct student s[])
{
// Loop to iterate over the students records //
for (i = 0; i < n; i++) {
printf("Name of student %s \n", s[i].name);
printf("Student roll number = %d \n",
s[i].rno); printf("Student fees status = %c \n", s[i].fees);
printf("Student number of days present = %f \n", s[i].days);
printf("Student attendence = %f \n", s[i].attend);
}}
```

```
// Function to Student Record//
void deleteRecord(struct student
s[]){ int a = 0;
printf("Enter the roll number of the student to delete it ");
scanf("%d", &a);
// Loop to iterate over the students records to delete the Data//
for (i = 0; i <= n; i++) {
// Condition to check the current student roll number is same as the user input roll
number//
if (s[i].rno == (a)) {
// Update record at ith index with (i + 1)th index //
for (j = i; j < n; j++) {
strcpy(s[j].name, s[j + 1].name);
s[j].rno = s[j + 1].rno;
s[j].fees = s[j + 1].fees;
s[j].days = s[j + 1].days;
s[j].attend = s[j + 1].attend;
}
printf("Student Record deleted");
}
}
}
```

```
// Function to print the student details of the
eligible students//
void eligibleStudents(struct student s[])
{
printf(" \n");
printf("Qualified student are = \n");
// Iterate over the list of the students records
// for (i = 0; i < n; i++) {
// Check for the eligibility of the student//
if (s[i].fees == money || 'B' == money) {
if (s[i].attend >= present) {
printf("Student name = %s \n",
s[i].name); printf("Student roll no. = %d \n",
s[i].rno);
printf(" Student fees = %c \n", s[i].fees);
printf(" Student attendence = %f \n", s[i].attend);
}
}
}
```

```
// Function to add the students record//
void add(struct student s[50])
{
printf("Enter the total ");
printf("number of working days \n");
scanf("%f", &tdays);
printf("Enter the number");
printf("of students \n");
scanf("%d", &n);
for (i = 0; i < n; i++)
{
printf("Student number %d \n", (i + 1));
printf("Enter the name of the student \n");
scanf("%s", s[i].name);
printf("Enter the roll number \n");
scanf(" %d", &s[i].rno);
printf("Enter the fees of the student 'P' for paid , 'N' for not paid \n");
scanf(" %c", &s[i].fees);
printf("Enter the number of days the student was present \n");
scanf("%f", &s[i].days);
s[i].attend = (s[i].days / tdays) * 100;
printf("student attendence = %f \n", s[i].attend);
}
execute();
}
```

```
/ Driver Code //
int main(){ printf("Welcome to Student database registration \n");
printf("Enter 0 to exit \n");
printf("Enter 1 to add student record \n");
scanf("%d", &option);
// Switch Statements //
switch (option)
{
case 0:
exit(0);
case 1:
add(s)
;
break;
default:
printf("Only enter 0 or 1");
execute();
}
return 0;
}
```

OUTPUT:

```
PS C:\cprogram.c> cd "c:\cprogram.c\" ; if ($?) { gcc SchoolExaminationSystem.c -o SchoolExaminationSystem } ; if ($?) { .\SchoolExaminationSystem }
Welcome to Student database registration
Enter 0 to exit
Enter 1 to add student record
1
Enter the total number of working days
150
Enter the numberof students
3
Student number 1
Enter the name of the student
Tanishq
Enter the roll number
2111003010086
Enter the fees of the student 'P' for paid , 'N' for not paid
P
Enter the number of days the student was present
145
student attendence = 96.666664
Student number 2
Enter the name of the student
Ajay
Enter the roll number
2111003010096
Enter the fees of the student 'P' for paid , 'N' for not paid
N
Enter the number of days the student was present
140
student attendence = 93.333336
Student number 3
Enter the name of the student
Akshit
Enter the roll number
2111003010088
Enter the fees of the student 'P' for paid , 'N' for not paid
P
Enter the number of days the student was present
135
student attendence = 90.000000
```

```
 Enter the serial number to select the option
 1. To show Eligible candidates
 2. To delete the student record
 3. To change the eligibility criteria
 4. Reset the eligibility criteria
 5. Print the list of all the student
 Enter 0 to exit
1

Qualified student are =
 Enter the serial number to select the option
 1. To show Eligible candidates
 2. To delete the student record
 3. To change the eligibility criteria
 4. Reset the eligibility criteria
 5. Print the list of all the student
 Enter 0 to exit
2
Enter the roll number of the student to delete it 2111003010086
Student Record deleted Enter the serial number to select the option
 1. To show Eligible candidates
 2. To delete the student record
 3. To change the eligibility criteria
 4. Reset the eligibility criteria
 5. Print the list of all the student
 Enter 0 to exit
1

Qualified student are =
 Enter the serial number to select the option
 1. To show Eligible candidates
 2. To delete the student record
 3. To change the eligibility criteria
 4. Reset the eligibility criteria
 5. Print the list of all the student
 Enter 0 to exit

3
Old Attendance required = 75.000000
 Enter the updated attendence required
60.00
fees status required was P
Enter the new fees status 'P' for paid 'N' for not paid and 'B' for both
Eligibility Criteria updated
 Enter the serial number to select the option
```

```
 Enter the serial number to select the option
 1. To show Eligible candidates
 2. To delete the student record
 3. To change the eligibility criteria
 4. Reset the eligibility criteria
 5. Print the list of all the student
 Enter 0 to exit
4
Eligibility creitria reset
 Enter the serial number to select the option
 1. To show Eligible candidates
 2. To delete the student record
 3. To change the eligibility criteria
 4. Reset the eligibility criteria
 5. Print the list of all the student
 Enter 0 to exit
5
Name of student Ajay
Student roll number = -2120899536
Student fees status = N
Student number of days present = 140.000000
Student attendence = 93.333336
Name of student Akshit
Student roll number = -2120899544
Student fees status = P
Student number of days present = 135.000000
Student attendence = 90.000000
Name of student
Student roll number = 0
Student fees status =
Student number of days present = 0.000000
Student attendence = 0.000000
 Enter the serial number to select the option
 1. To show Eligible candidates
 2. To delete the student record
 3. To change the eligibility criteria
 4. Reset the eligibility criteria
 5. Print the list of all the student
 Enter 0 to exit
0
PS C:\cprogram.c>
```