# **PROJECT**

### STUDENT LAPTOP SCHEME

## **REPORT**

In today's technologically advanced world, programming has become an indispensable skill, driving innovation and powering various industries. From mobile apps and websites to complex algorithms and artificial intelligence, the applications of programming are vast and everexpanding. Understanding programming fundamentals is the key to unlocking this realm of infinite possibilities.

In our completed project on the Student Laptop Scheme application process, we have harnessed the power of programming by employing structure arrays, strings, and functions. By leveraging these essential programming components, we have successfully developed a robust and efficient system that facilitates the application process for students seeking laptops. The use of structure arrays has allowed us to organize and manage the relevant information pertaining to each student, ensuring seamless data handling. Strings have played a vital role in capturing and manipulating textual data, enabling us to collect necessary details such as student names, addresses, and contact information. Moreover, the implementation of functions has provided modularization and code reusability, streamlining the development process and enhancing the overall functionality of our application. Through the judicious utilization of these programming fundamentals, we have created a user-friendly and efficient solution that simplifies the student laptop scheme application process, making it more accessible and inclusive for aspiring learners.

#### **Question:**

Students' Laptop Scheme Application Process: Explore the application process for the Students' Laptop Scheme and develop a program that simplifies the application procedure. Utilize user-defined functions, arrays, structures, and at least three built-in string functions to create an efficient system for laptop scheme applications. Format the output using <iomanip> functions for improved readability.

#### **PROGRAM**:

```
#include <iostream>
#include <iomanip>
#include <string>
#include <limits>
using namespace std;
// Structure to hold student information
struct Student {
  string name;
  int nicnumber;
  string city;
  string fathername;
  string university;
  int regNumber;
  string department;
  int semester;
  int Academic_performance;
};
```

```
void displaymenu()
{
cout<<setw(85)<<"*****************************
****"<<endl:
  cout<<endl;
  cout<<setw(85)<<" WELCOME TO THE STUDENT LAPTOP SCHEME
APPLICATION "<<endl;
  cout<<endl;
cout<<setw(85)<<"******************************
****"<<endl;
}
void getStudentDetails(Student& student) {
  cout << "Enter student name: ";
  getline(cin, student.name);
  cout << "Enter student NIC number: ";</pre>
  cin>>student.nicnumber;
  cout << "Enter your city of residence: ";
  getline(cin>>ws, student.city);
  cout << "Enter your father name: ";</pre>
  getline(cin >>ws, student.fathername);
   cout << "Enter your university name: ";</pre>
  getline(cin, student.university);
  cout << "Enter roll number: ";
  cin>> student.regNumber;
  cout << "Enter department: ";
```

```
getline(cin >>ws, student.department);
  cout << "Enter semester: ";
  cin >> student.semester;
  cout << "Enter Academic performance of the student(CGPA): ";
  cin >> student.Academic performance;
  cin.ignore(numeric_limits<streamsize>::max(), '\n');
}
void displayStudentDetails(const Student& student) {
 cout<<endl;
 cout<<setw(70)<<"___STUDENT DETAILS____"<<endl;
  cout<<endl;
 cout << "NAME: " << student.name << endl;</pre>
  cout << "STUDENT NIC NUMBER: " << student.nicnumber << endl;</pre>
  cout << "STUDENT CITY: " << student.city << endl;
  cout << "FATHER NAME: " << student.fathername << endl;</pre>
 cout << "UNIVERSITY NAME: " << student.university << endl;</pre>
 cout << "REGISTRATION NUMBER: " << student.regNumber << endl;</pre>
  cout << "STUDENT DEPARTMENT: " << student.department << endl;</pre>
  cout << "SEMESTER: " << student.semester << endl;</pre>
}
bool isEligible(const Student& student) {
  if (student.department == "CS" && student.Academic_performance >= 3) {
```

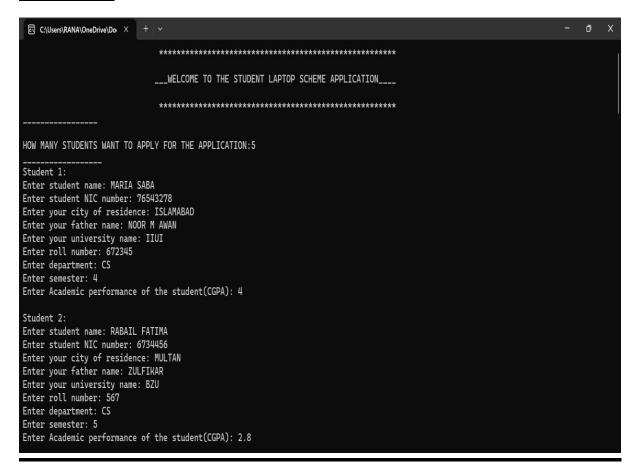
```
return true; // Student is eligible
 } else {
   cout << "Student is ineligible due to one or more of the following
reasons:\n";
    if (student.department != "CS") {
     cout << "- Student does not belong to the CS department.\n";
   }
   if (student.Academic performance < 3) {
     cout << "- Student has an Academic performance of less than 3
CGPA.\n";
   }
   return false; // Student is not eligible
 }
}
void submitApplication(const Student& student) {
cout<<endl;
 cout<<setw(70)<<"__APPLICATION SUBMITTED SUCCESSFULLY____"<<endl;
 cout<<endl;
}
int main() {
 const int MAX STUDENTS = 100;
 Student students[MAX_STUDENTS];
```

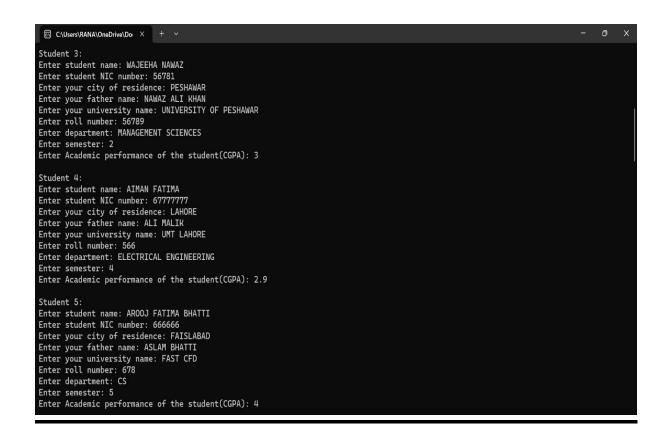
```
int numStudents;
 displaymenu();
 cout<<"_____"<<endl;
                                "<<endl;
 cout<<"
 cout<<"HOW MANY STUDENTS WANT TO APPLY FOR THE APPLICATION:";
 cin >> numStudents;
 cout<<"_____";
 cin.ignore(numeric limits<streamsize>::max(), '\n'); // Clear input buffer
 // total number of students allowed.
 if (numStudents > MAX STUDENTS) {
cout<<setw(70)<<"
   cout<<setw(70)<<" SORRY! YOUR APPLICATION CANNOT BE
PROCESSED ";
   cout<<setw(70)<< " MAXIMUM NUMBER OF STUDENTS ALLOWED
IS_____ " << MAX_STUDENTS << endl;
   return 1;
 }
 // Get details for each student
 for (int i = 0; i < numStudents; i++) {
   cout << "\nStudent " << i + 1 << ":" << endl;
   getStudentDetails(students[i]);
 }
```

```
// Display details and submit applications for eligible students
 cout<<setw(75)<<" "<<endl;
 cout<<setw(75)<< "**** APPLICATION RESULTS***" <<endl;
 cout<<setw(75)<<"_____"<<endl;
 for (int i = 0; i < numStudents; i++) {
   cout << "\nStudent " << i + 1 << ":" << endl;
   displayStudentDetails(students[i]);
   if (isEligible(students[i])) {
    cout<<endl;
    cout<<endl;
cout<<setw(70)<<"_____
                                                     "<<e
ndl;
    cout<<setw(70)<<
cout<<setw(70)<<" YOU ARE ELLIGIBLE FOR THE LAPTOP
APPLICATION "<<endl;
cout<<setw(70)<<"_____
                                                      "<<
endl;
    cout<<endl;
    cout<<endl;
    submitApplication(students[i]);
   } else {
    cout<<endl;
    cout<<endl;
```

}

#### **Output**







© C:\Users\RANA\OneDrive\Do: X +		ð	Χ
Student 2:			
Student 2:	*******		
	STUDENT DETAILS		
	*********		
NAME: RABAIL FATIMA STUDENT NIC NUMBER: 6734456 STUDENT CITY: MULTAN FATHER NAME: ZULFIKAR UNIVERSITY NAME: BZU REGISTRATION NUMBER: 567 STUDENT DEPARTMENT: CS SEMESTER: 5 Student is ineligible due to - Student has an Academic per	one or more of the following reasons: formance of less than 3 CGPA.		
	WE ARE SORRY UL ARE NOT ELLIGIBLE FOR THE APPLICATION		
***	***************************************		

C:\Users\RANA\OneDrive\Do: X			ð	Χ		
Student 3:						
	**********					
	STUDENT DETAILS					
	*********					
NAME: WAJEEHA NAWAZ						
STUDENT NIC NUMBER: 56781						
STUDENT CITY: PESHAWAR						
FATHER NAME: NAWAZ ALI KHA	N					
UNIVERSITY NAME: UNIVERSIT	Y OF PESHAWAR					
REGISTRATION NUMBER: 56789						
STUDENT DEPARTMENT: MANAGE	MENT SCIENCES					
SEMESTER: 2						
Student is ineligible due	to one or more of the following reasons:					
	- Student does not belong to the CS department.					
	******************************					
	WE ARE SORRY					
-	_YOU ARE NOT ELLIGIBLE FOR THE APPLICATION					
	*******************************					

©:\Users\RANA\OneDrive\Do:\X		ð	Χ
Student 4:			
Seaucife 4.	**********		
	STUDENT DETAILS		
	********		
- Student does not belong	E ICAL ENGINEERING to one or more of the following reasons:		
-	**************************************		
-			

© C:\Users\RANA\OneDrive\Do⊢ X		0	Χ
Student 5:			
	**********		
	STUDENT DETAILS		
	*********		
NAME: AROOJ FATIMA BHATTI STUDENT NIC NUMBER: 666666 STUDENT CITY: FAISLABAD FATHER NAME: ASLAM BHATTI UNIVERSITY NAME: FAST CFD REGISTRATION NUMBER: 678 STUDENT DEPARTMENT: CS SEMESTER: 5			
-	**************************************		
_Y	OU ARE ELLIGIBLE FOR THE LAPTOP APPLICATION_		
<u></u>	*******************************		
	APPLICATION SUBMITTED SUCCESSFULLY		
	************************		1