|  |  |  |
| --- | --- | --- |
| **PROJECT REPORT** | **December 19**  2017 | |
| **SUBMITTED TO: SIR SUBMITTED BY: EHTESHAM SAEED (160503) ASIM JAVED (160515) DANISH ALI (160) BABAR ALI (160910) DATED: 19-12-2017** | |  |

****

**Student Management System**

**Student Management System:** The use of computer is emerging in every field of life. Nearly a computer simulation of every physical phenomena is prepared. In the past, it was hard to keep the record of students. Now a days, computer programs are developed which stores the records and processes them efficiently.

This project is also one of them and it is developed in C++ programming language.

The special thing about this Student Management System Project is that it uses **LINKED LIST.**

.

**Preference of Linked List :** Following are the advantage of linked list over arrays.

* It's easier to store data of different sizes in a linked list. An array assumes every element is exactly the same size.
* It's easier for a linked list to grow organically. An array's size needs to be known ahead of time, or re-created when it needs to grow.
* Shuffling a linked list is just a matter of changing what points to what. Shuffling an array is more complicated and/or takes more memory.

**Functions Used in the Program:**   
 The following functions are used in the program:

**void insert();** This function is called in the beginning of the game. It starts the creation of the node asking basic student info from the user in our case

**First Name, Second Name, Roll ID, Date of birth ,CGPA.**

**void edit();** If the user wants to edit the data he has entered he can enter the roll id of the student the function will traverse through the nodes in search of the roll id and when found it will ask the user to over right the data..

**void delete();**

If the user wants to delete the data he can use the delete function from the menu which first asks for roll id and traverses in search of id through nodes and when found erase the node.

**void display();**

This function displays all the nodes.

**void exit();**

This function helps to exit the program.

**PROGRAM**

#include<iostream>

#include <cstdlib>

using namespace std;

string x,y;

double z,date,month,year,numberofstudents,b;

int choice,eid,fid,delid;

class list{

private:

struct node{

string firstName,lastName;

int id,dob[3];

double gpa;

node \*next;

};

public:

node \*head;

list(){

head=NULL;

}

~list(){

node \*t;

t=head;

while(head!=NULL){

head=head->next;

delete t;

}

}

void insert(string x,string y,int z,double b,int date,int month,int year)

{

node \*ptr = new node;

ptr->firstName=x; // adding data

ptr->lastName=y;

ptr->id=z;

ptr->dob[0]=date;

ptr->dob[1]=month;

ptr->dob[2]=year;

ptr->gpa=b;

ptr->next = NULL; // setting it to NULL

if (head == NULL)

{

head = ptr;

}

else

{

node \*temp;

temp = head;

while(temp->next != NULL)

{

temp = temp->next;

}

temp ->next = ptr;

}

}

void edit()

{

cout << "Please Enter Student's ID: ";

comehere:

cin >> eid;

system("CLS");

node \*p = new node;

// If the list is empty, do nothing

if (!head)

{

cout<< "NO RECORD FOUND" << endl;

return;

}

p = head;

while (p!=NULL)

{

if(p->id == eid)

{

cout << "Enter First Name:" << endl;

cin >> x;

p->firstName=x;

cout << "Enter Last Name:" << endl;

cin >> y;

p->lastName=y;

cout << endl;

cout << "Enter ID:" << endl;

cin >> z;

p->id=z;

cout << endl;

cout << "Enter Date of Birth:" << endl;

cout << "Day: " << endl;

cin >> date;

while (date < 1 || date > 31)

{

cout << "Invalid Date. Enter again: " << endl;

cin >> date;

}

p->dob[0]=date;

cout << "Month:" << endl;

cin >> month;

while (month < 1 || month > 12)

{

cout << "Invalid Month. Enter again: " << endl;

cin >> month;

}

p->dob[1]=month;

cout << "Year:" << endl;

cin >> year;

while (year < 1900 || year > 2017)

{

cout << "Invalid Year. Enter again: " << endl;

cin >> year;

}

p->dob[2]=year;

cout << endl;

cout << "Enter CGPA:" << endl;

cin >> b;

while (b < 1 || b>4)

{

cout << "Invalid GPA. Enter again: ";

cin >> b;

}

p->gpa=b;

cout << endl;

system("CLS");

return;

}

p=p->next;

}

cout << "INVALID ID ENTERED" << endl;

cout << "Enter Again: ";

goto comehere;

}

void search()

{

cout << "Please Enter Student's ID: ";

comehere:

cin >> fid;

system("CLS");

node \*p = new node;

// If the list is empty, do nothing

if (!head)

{

cout<< "NO RECORD FOUND" << endl;

return;

}

p = head;

while(p!=NULL)

{

if(p->id == fid)

{

cout << "DISPLAYING THE SEARCHED DATA"<< endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "Student's Name:"<<"\t" << p->firstName << " " << p->lastName << endl;

cout << "Student's ID: "<< p->id<< endl;

cout << "Student's CGPA: " << p->gpa << endl;

cout << "Student's Date of Birth: ";

for(int l=0; l<3; l++)

{

if (l == 2)

{

cout << p->dob[l] << endl;

}

else

{

cout << p->dob[l];

cout << "-";

}

}

return;

}

p=p->next;

}

cout << "INVALID ID ENTERED" << endl;

cout << "Enter Again: ";

goto comehere;

}

void del()

{

cout << "ENTER THE STUDENT ID FOR DELETION" << endl;

cin >> delid;

node \*nodePtr, \*previousNode;

// If the list is empty, do nothing

if (!head)

{

cout<< "NO RECORD FOUND" << endl;

return;

}

// Determine if the first node is the one.

if (head->id == delid)

{

nodePtr = head->next;

delete head;

head = nodePtr;

}

else

{

// Initialize nodePtr to head of list

nodePtr = head;

// Skip all nodes whose value member is not equal to num.

while (nodePtr != NULL && nodePtr->id != delid)

{

previousNode = nodePtr;

nodePtr = nodePtr->next;

}

// Link the previous node to the node after

// nodePtr, then delete nodePtr.

previousNode->next = nodePtr->next;

delete nodePtr;

}

}

void display()

{

if (!head)

{

cout<< "NO RECORD FOUND" << endl;

return;

}

node \*t=head;

int i=1;

while(t!=NULL)

{

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "DATA OF STUDENT: " << i <<endl;

cout << "Student's Name: " << t->firstName << " " << t->lastName << endl;

cout << "Student's ID: " << t->id << endl;

cout << "Students's Date of Birth: ";

for(int x=0; x<3; x++)

{

if (x == 2)

{

cout << t->dob[x];

}

else

{

cout << t->dob[x];

cout << "-";

}

}

cout << endl;

cout << "Student's CGPA: " << t->gpa << endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << endl;

t=t->next;

i++;

}

}

};

//the menu is here

void menu(list &l)

{

comehere:

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << endl;

cout << " PRESS 1 TO ADD RECORD OF A STUDENT " << endl;

cout << " PRESS 2 TO EDIT THE RECORD OF A STUDENT " << endl;

cout << " PRESS 3 TO SEARCH THE RECORD OF A STUDENT " << endl;

cout << " PRESS 4 TO DELETE THE RECORD OF A STUDENT " << endl;

cout << " PRESS 5 TO DISPLAY THE RECORD OF A STUDENT " << endl;

cout << " PRESS 6 TO EXIT THE PROGRAM " << endl;

cout << endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ " << endl;

cout <<endl;

cout << " CHOOSE FROM MENU \nENTER YOUR OPTION : "; cin >> choice;

while (choice < 1 || choice > 6)

{

cout << "Invalid Optopn. Enter again: ";

cin >> choice;

}

switch(choice)

{

case 1:

cout << endl;

cout <<"How many student's data you want to enter?" << endl;

cin >> numberofstudents;

cout << endl;

system("CLS");

for(int i=0; i<numberofstudents; i++)

{ cout << endl;

cout << "DATA ENTRY OF STUDENT "<< i+1 <<endl;

cout << "Enter First Name:" << endl;

cin >> x;

cout << "Enter Last Name:" << endl;

cin >> y;

cout << endl;

cout << "Enter ID:" << endl;

cin >> z;

cout << endl;

cout << "Enter Date of Birth:" << endl;

cout << "Day: " << endl;

cin >> date;

while (date < 1 || date > 31)

{

cout << "Invalid Date. Enter again: " << endl;

cin >> date;

}

cout << "Month:" << endl;

cin >> month;

while (month < 1 || month > 12)

{

cout << "Invalid Month. Enter again: " << endl;

cin >> month;

}

cout << "Year:" << endl;

cin >> year;

while (year < 1900 || year > 2017)

{

cout << "Invalid Year. Enter again: " << endl;

cin >> year;

}

cout << endl;

cout << "Enter CGPA:" << endl;

cin >> b;

while (b < 1 || b>4)

{

cout << "Invalid GPA. Enter again: ";

cin >> b;

}

cout << endl;

l.insert(x,y,z,b,date,month,year);

system("CLS");

}

goto comehere;

case 2:

system("CLS");

l.edit();

goto comehere;

break;

case 3:

system("CLS");

l.search();

goto comehere;

break;

case 4:

system("CLS");

l.del();

goto comehere;

case 5:

system("CLS");

l.display();

goto comehere;

case 6:

system("CLS");

cout<<"YOU HAVE SELECTED TO EXIT THE PROGRAM";

exit(1);

}

}

int main()

{

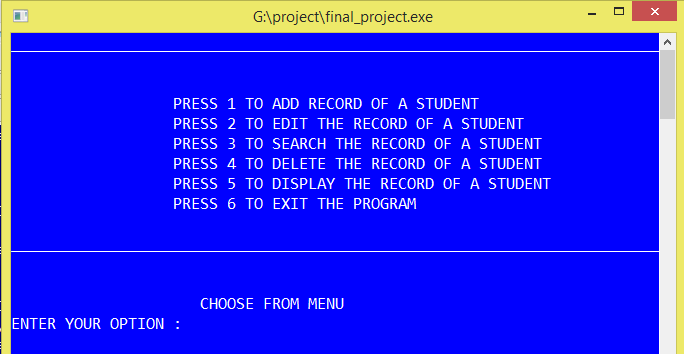
list c;

menu(c);

}

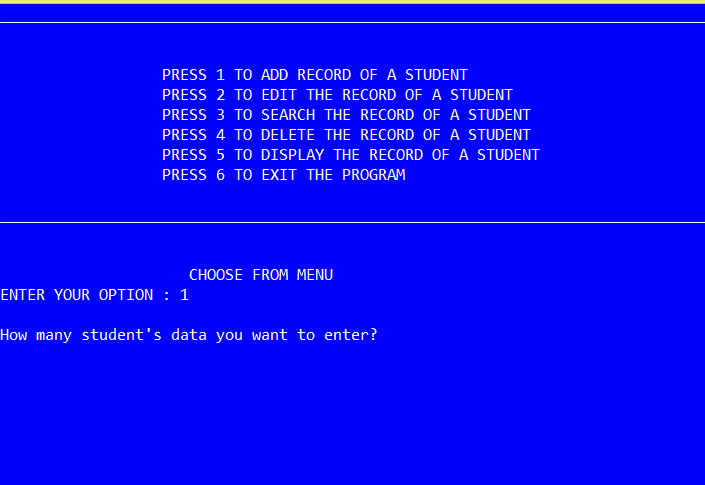
**OUTPUT:**

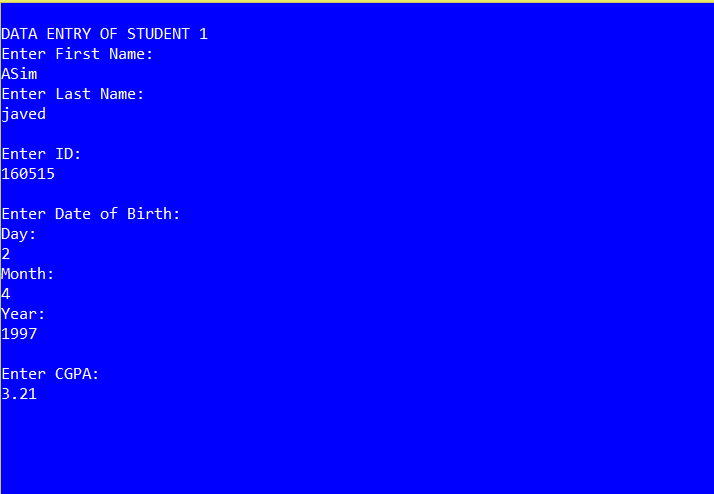
**The Menu:**

****

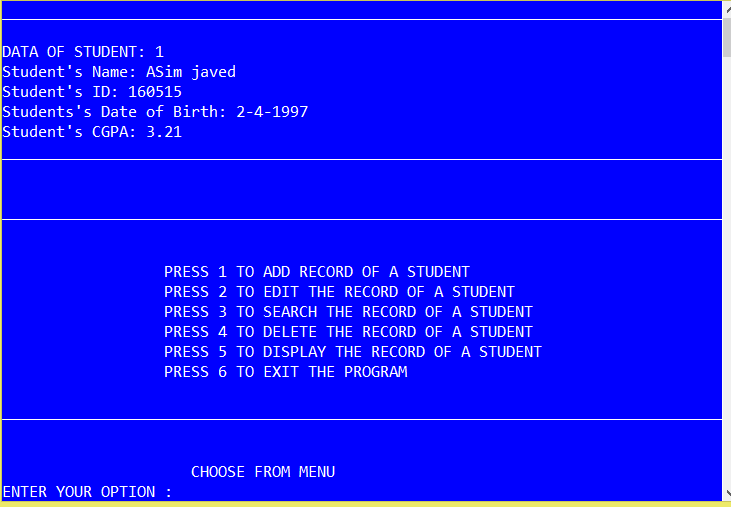
**INSERT:**

Asking the number of students from the user,which he want to enter for insertion.

****

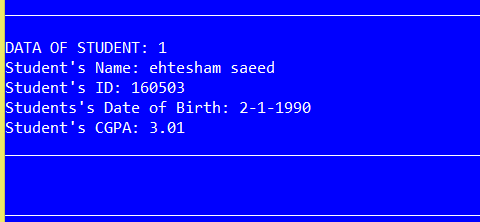
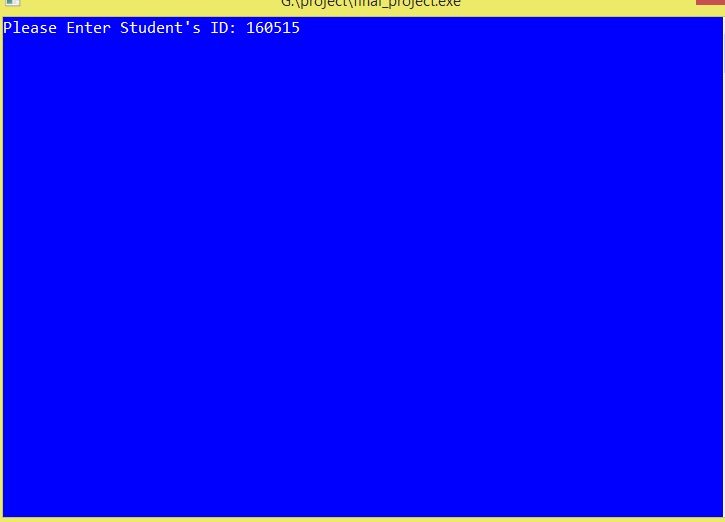
After data entry from the user**.**

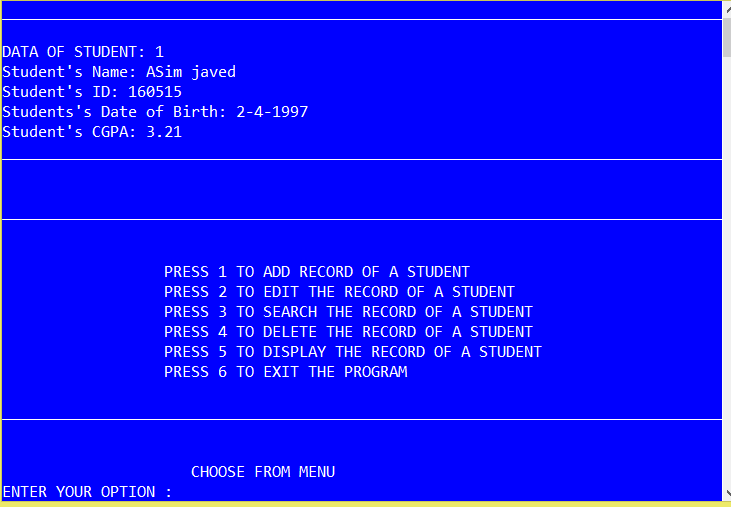
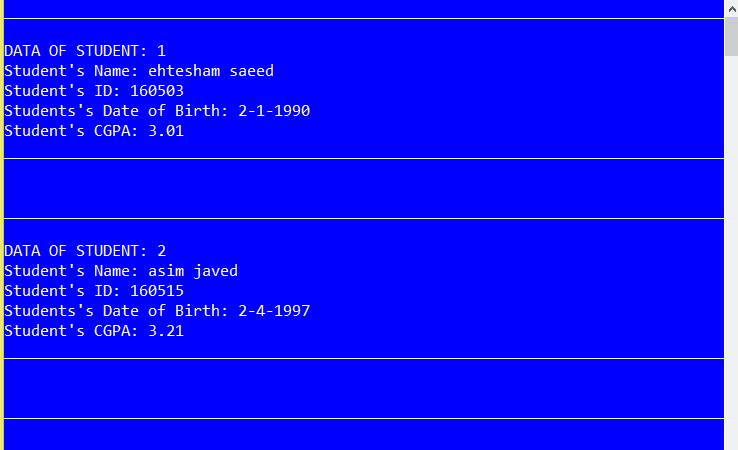
**DISPLAY:**

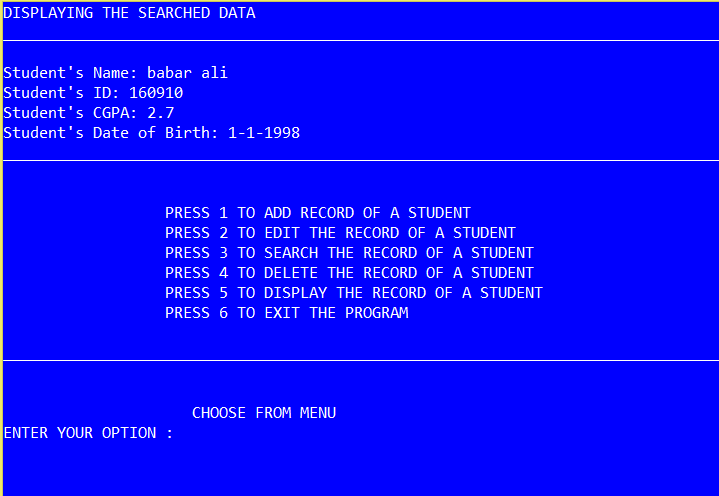
****

**EDIT:**

Data is over writtern.

****

**DELETION:**

**SEARCH FUNCTION:**