```cpp
//*********************************************************************************************************************************
//                    HW 1
//                    Name:  Edgar Zapata
//                    3013 Date: Date of Submission (9/22/2021)
//*********************************************************************************************************************************
//                  Program:
//                     Using the STL list libray this program will make a database
//                     to list records of users inputs.
//                     Will be  using the librays set functions to pop elements
//                     from the head and tail. push elements to the rear.
//                     Display the the datebase with the contents
//
//*********************************************************************************************************************************

//*********************************************************************************************************************************
//                 FOR ALL FUNCTIONS function Name:
//                      struct student:
//                      created the data for the linkedlist to be adding on the elements
//                      for the datebase
//
//*********************************************************************************************************************************
//*********************************************************************************************************************************
//          How to run program:
//              Use visual studio code or visual studio and run the code from there.
//              Or use a termnial and c++ or g++ the program. Then ./a.out to run
//
//*********************************************************************************************************************************




#include <iostream>
#include <list>                                                          // open the list library
#include <string>
using namespace std;
                                                                         // struct to make the elements in the linked list easier to pass
struct student
{

    string first_name;                                                   // user first name
    string last_name;                                                    // user last name
    char gender;                                                         // gender
    int id;                                                               // user id
    int age;                                                              // user age

                                                                          // default constructor to be able to pass the elements
    student(string fname, string lname, char g, int student_age, int student_id)
    {
        id = student_id;                                                  // seting these values to be equal to varvable list in the struct
        first_name = fname;
        last_name = lname;
        gender = g;
        age = student_age;
    }
};

int main()
{

    list<student> List_1;                                                 // creating the linked list with the stl list library
    list<student>::iterator list1;                                        // iterator list1 to be able to point to the elements or access list ones elements

    string fname;
    string lname;
    char gender;
    int id;
    int age;
                                                                          // giving the user to input the first element
    cout << "enter the first  name" << endl;
    cin >> fname;
    cout << "enter the last  name" << endl;
    cin >> lname;
    cout << "enter the id" << endl;
    cin >> id;
    cout << "enter age" << endl;
    cin >> age;
    cout << "enter the gender" << endl;
    cin >> gender;

    List_1.push_back(student(fname, lname, gender, age, id));            // using pushback to place the values from the rear

    while (true)                                                          // while loop to keep looping for the user to able to add more users
    {
        cout << "do you want to enter more. Enter Y or y for yes?" << endl;
        char input_new;
        cin >> input_new;
        if (input_new == 'Y' || input_new == 'y')                         // if statment to active the user inputs
        {
            cout << "enter the first  name" << endl;
            cin >> fname;
            cout << "enter the last  name" << endl;
            cin >> lname;
            cout << "enter the id" << endl;
            cin >> id;
            cout << "enter age" << endl;
            cin >> age;
            cout << "enter the gender" << endl;
            cin >> gender;
            List_1.push_back(student(fname, lname, gender, age, id));
            continue;                                                   // to keep looping the while loop untill break
        }
        else
        {
            break;                                                      // kills the while loop
        }
    }

    cout << endl;
    for (list1 = List_1.begin(); list1 != List_1.end(); list1++)        // for loop to display the elements in  list 1
                                                                        // the for loop is traversing the linked list to the end
    {
        int id = list1->id;                                             // this for accessing the data from the struct will be doing this for all the data types
        int age = list1->age;
        char gender = list1->gender;
        string fname = list1->first_name;
        string lname = list1->last_name;

        cout << fname << " " << lname << " - "                          // couts out the first name, last name , and id of the users
             << "id:" << id;
        cout << endl;
    }

    cout << endl;

    cout << "Do you want to remove the head enter Y or y for yes?" << endl;

    char input;
    cin >> input;

    if (input == 'Y' || input == 'y')                                   // if statment to evaulate the user input
    {
        if (List_1.empty())                                             // shows the list is empty for safety purpose to not error the program
        {
            cout << "empty list can't pop anything" << endl;
        }
        else
        {
            List_1.pop_front();                                         // removes the head of the linked list
        }
    }
```

```cpp
    else                                                                // if user does not want to remove the head
    {
        cout << "Head not popped" << endl;
    }

    cout << endl;

    for (list1 = List_1.begin(); list1 != List_1.end(); list1++)        // outputs the new list with the removed head
    {
        int id = list1->id;
        int age = list1->age;
        char gender = list1->gender;
        string fname = list1->first_name;
        string lname = list1->last_name;

        cout << fname << " " << lname << " - "
             << "id:" << id;
        cout << endl;
    }
    cout << endl;

    cout << "Do you want to remove the tail enter Y or y for yes" << endl;
    char input_2;
    cin >> input_2;
    cout << endl;

    if (input_2 == 'Y' || input_2 == 'y')
    {

        if (List_1.empty())                                             // shows the list is empty for safety purpose to not error the program
        {
            cout << "empty list can't pop anything" << endl;
        }
        else
        {
            List_1.pop_back();                                          // pops the element from the rear
        }
    }
    else
    {
        cout << "Tail not popped" << endl;                              // if user choose not to pop the rear
    }
    cout << endl;

    for (list1 = List_1.begin(); list1 != List_1.end(); list1++)        // displays new list with tailed being removed
    {
        int id = list1->id;
        int age = list1->age;
        char gender = list1->gender;
        string fname = list1->first_name;
        string lname = list1->last_name;

        cout << fname << " " << lname << " - "
             << "id:" << id;
        cout << endl;
    }

    cout << endl;

    while (true)                                                                    // while to ask the user if wants to add more user
    {
        cout << "Do you want to enter more. Enter Y or y for yes?" << endl;
        char input_new;
        cin >> input_new;
        if (input_new == 'Y' || input_new == 'y')
        {
            cout << "enter the first  name" << endl;
            cin >> fname;
            cout << "enter the last  name" << endl;
            cin >> lname;
            cout << "enter the id" << endl;
            cin >> id;
            cout << "enter age" << endl;
            cin >> age;
            cout << "enter the gender" << endl;
            cin >> gender;
            List_1.push_back(student(fname, lname, gender, age, id));
            continue;
        }
        else
        {
            break;
        }
    }

    cout << endl;

                                                                                    // display just the first name
    for (list1 = List_1.begin(); list1 != List_1.end(); list1++)
    {
        int id = list1->id;
        int age = list1->age;
        char gender = list1->gender;
        string fname = list1->first_name;
        string lname = list1->last_name;

        cout << fname;
        cout << endl;
    }

    return 0;
}
```