|  |
| --- |
| Department of Computer & Software Engineering  **Bahria University Karachi Campus**  13 National Stadium Road |

|  |
| --- |
| Software Design and Architecture |
| Student Course Management System |
| |  |  | | --- | --- | | **Submitted To** | **Engr. Rehan Baig** | | **Students** |  | | Name: MOEEZ AHMED SHAH | REG # 57266 | |  |  | |

**Table of Contents**

[01 Introduction: 2](#_Toc451639159)

[02 Goal 2](#_Toc451639160)

[03 Existing solutions / Projects 3](#_Toc451639161)

[04 Proposed Solution 5](#_Toc451639162)

[05 Work Flow Diagram 7](#_Toc451639163)

[06 ERD 9](#_Toc451639164)

[07 Database Queries 10](#_Toc451639165)

[07.1 Queries 10](#_Toc451639166)

[08 GUI Specification 18](#_Toc451639170)

[09 Conclusion 26](#_Toc451639171)

[10 References 27](#_Toc451639172)

**Git repository:** <https://github.com/moeez-ahmed/shah>

**Moeez commit ID:** 74b5b982eaff915043b6ef14f333265a5e6252f3

**Asad commit ID:** 2d1eea56e262b417594aadb222e199ba94d0f43b

**Obaid commit ID:** 2a8622c473f668200330325c0cd193e3851f8574

**Introduction:**

The use of tools to facilitate teaching is not new. For centuries educators have used blackboards, slates, charts, flash cards, books, and other teaching aids. The advent of electricity brought new tools such as slide projectors, phonographs, mimeographs, and movie projectors. In more recent times, tape recording technology has enabled presentation of audio and video content and photocopiers have facilitated bulk copying of printed text.

In the 1980s, educators were quick to adopt use of the nascent Internet. Electronic mail provided a speedy and convenient way to link teachers and students via mailing lists, and File Transfer Protocol (FTP) permitted rapid exchange of data documents.

In the 1990s the growth of the Worldwide Web (WWW) furnished teachers and students with an even more efficient way to broadcast announcements and course materials. Web-based bulletin boards facilitated teacher-student interaction.

Course Management Systems (CMS) are software packages that bring together numerous Internet tools and allow them to be accessed via a simple and convenient user interface. Such systems are also referred to as Virtual Learning Environments (VLE) or Learning Management Systems (LMS).

**Goal:**

These core objectives were created to ensure that each project initiative was in line with the needs of our Student Course Management System and the educators that use it.

* **Usability:** The entire installation system should be easy to use for educators of intermediate computer literacy. All parts of the system must provide proper feedback and information as to the progress of the installation.
* **Scalability:** The installer should be flexible to allow the addition of new features and configurable modules. The system should not depend on file layout or the existence of specific files.
* **Compatibility:** The installer should run successfully for a myriad of hosts, whether dedicated or shared.

**Existing Solution / Projects:**

A number of complete, open source Course management systems are available on the internet. One is called “Moodle.” The website can be found at the URL http://www.moodle.org. This website explains the four different categories of pedagogical principles which help educators create effective online learning communities.

Moodle is difficult to install and configure. In fact, the instructions for installation is seven pages of text! It requires the configuration of Apache web server, MySQL, and PHP to make the program work. Educators typically do not have the skillsets involved to configure such complex applications

**Proposed Solution:**

We will be developing a Student Course management system we will be using C# programming language for the front-end development and SQL for back-end development. Our Student Course management system will have the following tables present in its database that will be updated from time to time whenever there is any upgradation in the system by the admin.

Our Student Course management system will have interactive and user-friendly interface. It would be really simple to use and easy to learn and remember. It will fulfill all the advance needs of a Student Course management system.

After significant observation and study, a formalized installation procedure for the Student Course Management System was created. Utilizing agile software techniques and common practice human-computer-interaction procedures for testing, the project team accomplished its goal of establishing a usable, scalable, and compatible solution.

The installation package should provide for repeatable, understandable, and complete installation routines. Testing indicates that educators of average computer literacy can in fact install the Student Course Management System toolset easily. Now, educators can continue using Student Course Management System as they begin teaching new courses. This results in higher educator productivity and satisfaction with the system.

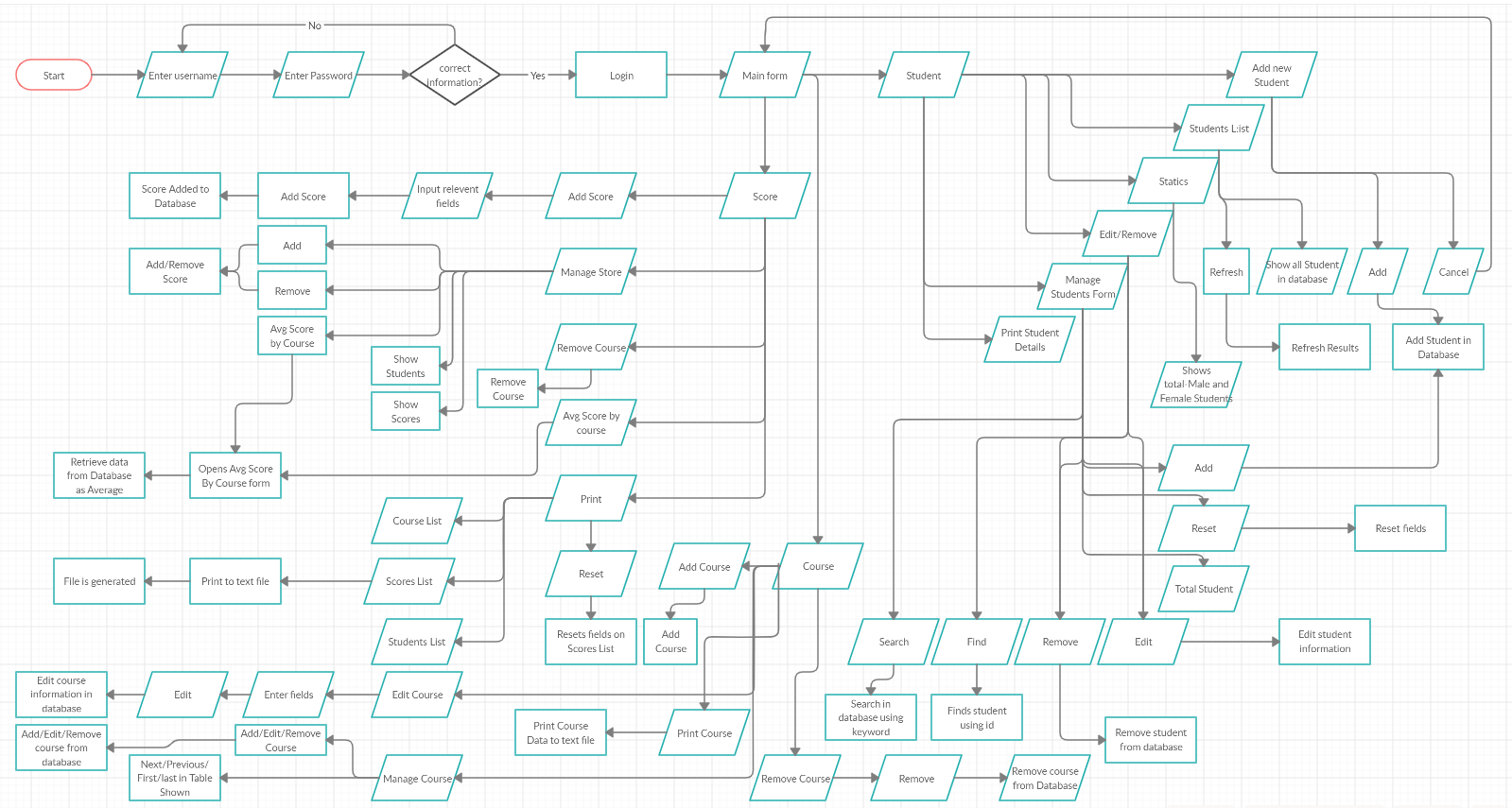
**Architecture Style:**

For this Student course management system, we are using 3 tier architectures. The presentation layer that contains our GUI and the data logic layer that consists of C#. The data storage layer consists of Database that is SQL in our case. We have used intense database terminologies in order to make our project work.

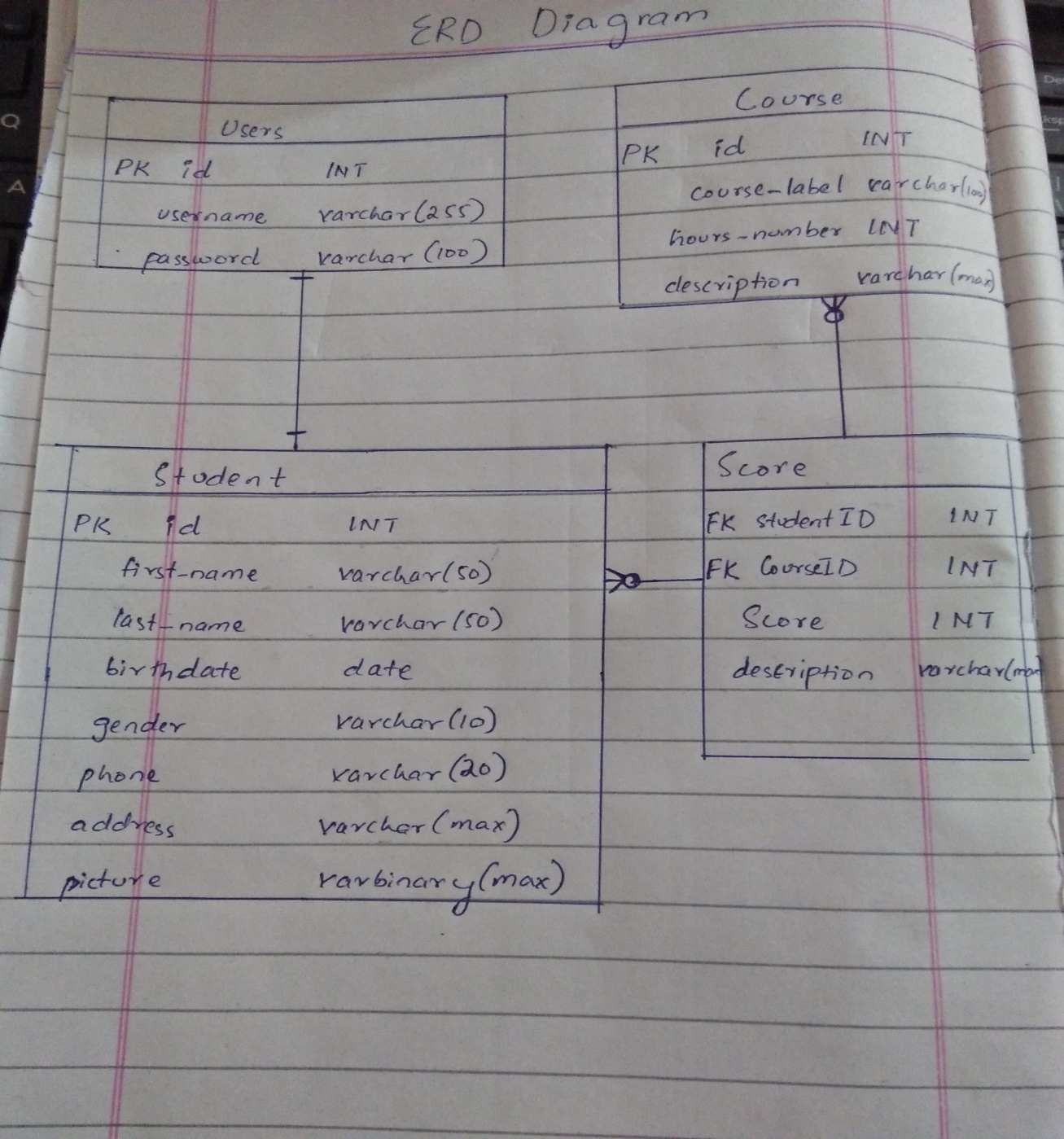
**Design Pattern:**

For our project we are using factory design pattern. As we have our management system that has a factory like structure encapsulated inside it.

**Work Flow Diagram:**



**ERD:**

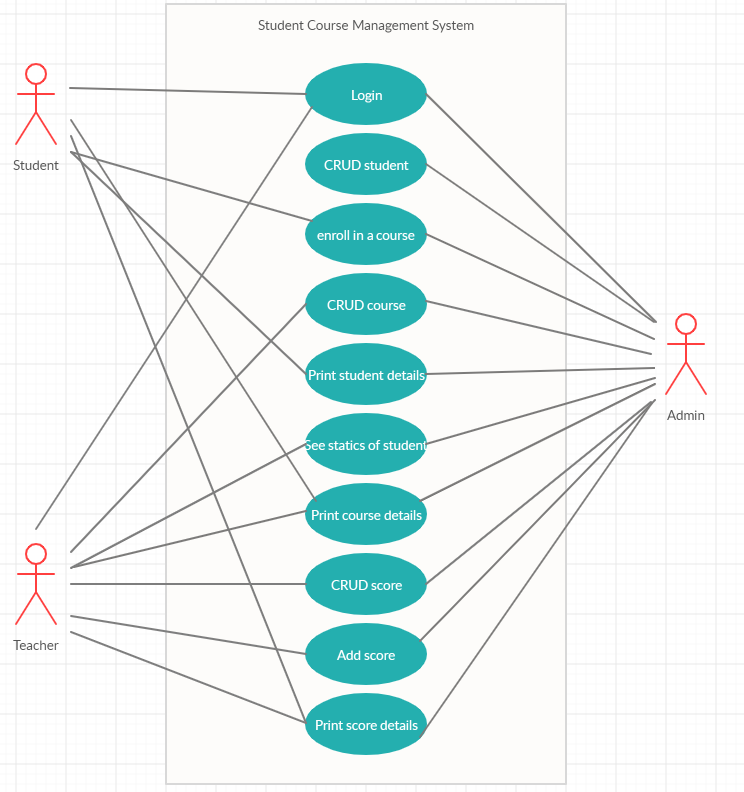
****

The relationship between Users and student is 1 to 1.

The relationship between Course and Student is many to many. That means many student can enroll in many courses and similarly many courses can be done by 1 and many student.

The score is a table formed as many to many relationship and that`s why holds ID of both Student and course as foreign keys.

**Use case Diagram:**



**Database Queries:**

* **Queries:**

create table users(

id INT IDENTITY(1,1) NOT NULL,

username varchar(255) NOT NULL,

password varchar(100) NOT NULL,

PRIMARY KEY(id)

);

INSERT INTO users(username, password)

VALUES('moeez', 'moeez');

INSERT INTO users(username, password)

VALUES('asad', 'asad');

create table student(

id INT IDENTITY(1,1) NOT NULL,

first\_name varchar(50) NOT NULL,

last\_name varchar(50) NOT NULL,

birthdate date NOT NULL,

gender varchar(10) NOT NULL,

phone varchar(20),

address varchar(max),

picture varbinary(max),

PRIMARY KEY(id)

);

create table course(

id INT IDENTITY(1,1) NOT NULL,

course\_label varchar(100) NOT NULL,

hours\_number int NOT NULL,

description varchar(max) NOT NULL,

PRIMARY KEY(id)

);

create table score(

studentId INT NOT NULL,

courseId INT NOT NULL,

score int NOT NULL,

description varchar(max) NOT NULL,

FOREIGN KEY(studentId) References student(id),

FOREIGN KEY(courseId) References course(id)

);

INSERT INTO score(studentId, courseId, score, description) VALUES(@SID, @CID, @scr, @desc)

**Query for average of score in each course:**

SELECT course.course\_label, AVG(score.score) as 'Average\_Score' FROM course,score WHERE course.id = score.courseId GROUP BY course.course\_label

**Inner join for foreign relations in score:**

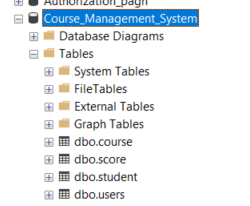
SELECT score.studentId, student.first\_name, student.last\_name, score.courseId, course.course\_label, score.score

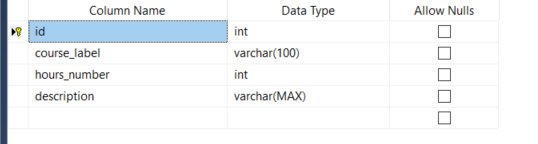
FROM student INNER JOIN score ON student.id = score.studentId INNER JOIN course ON score.courseId = course.id

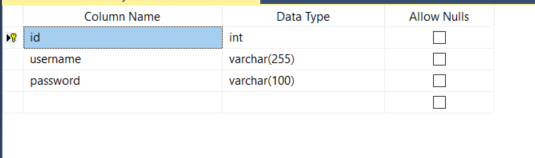
**For Deleting Score:**

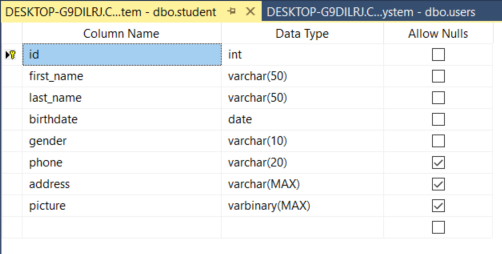
DELETE FROM score WHERE studentId = @SID AND courseId = @CID

SELECT id, first\_name, last\_name, birthdate FROM student

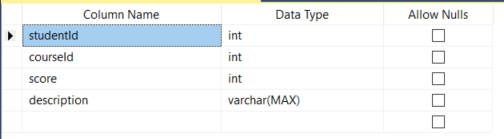


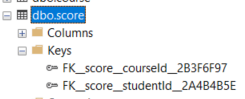






**Score Table:**



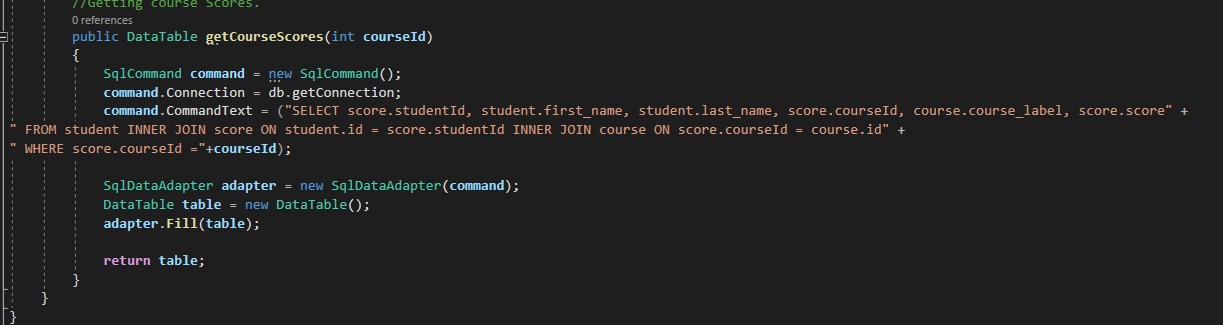


**getting course scores using course ID:**

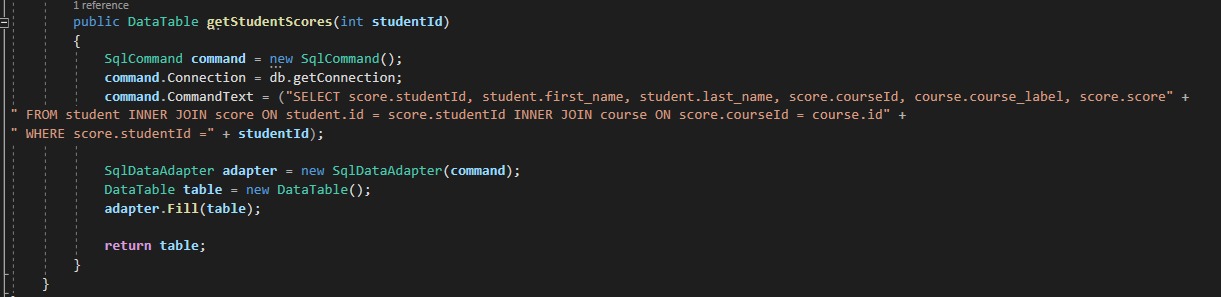
SELECT score.studentId, student.first\_name, student.last\_name, score.courseId, course.course\_label, score.score

FROM student INNER JOIN score ON student.id = score.studentId INNER JOIN course ON score.courseId = course.id

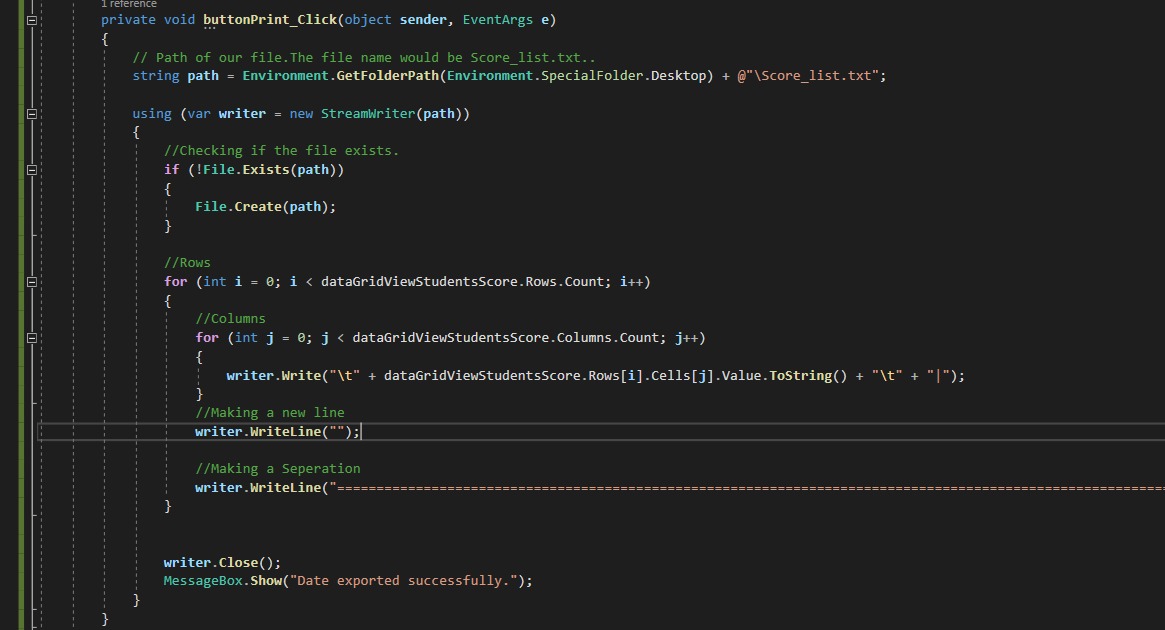
WHERE score.courseId = @CID



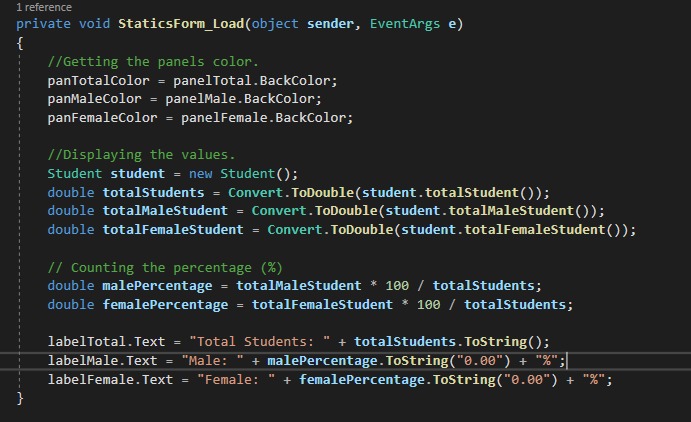
**using studentId:**

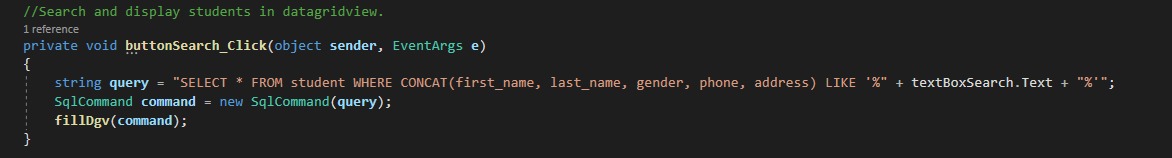


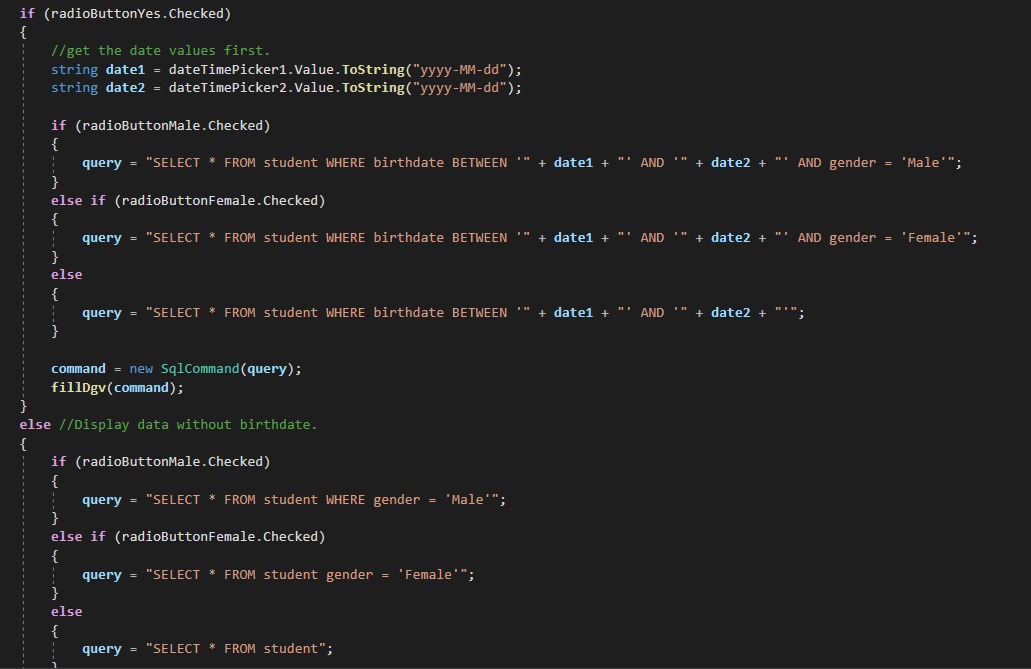
**printing data to file:**

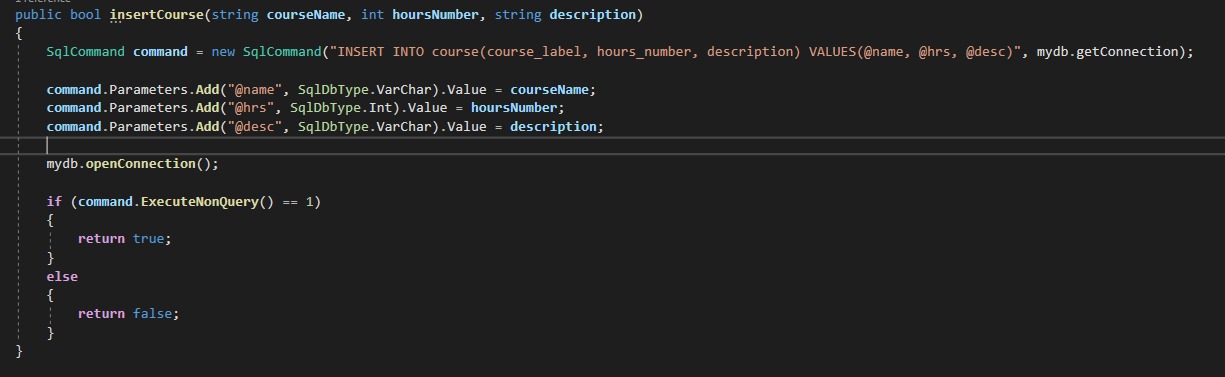
****

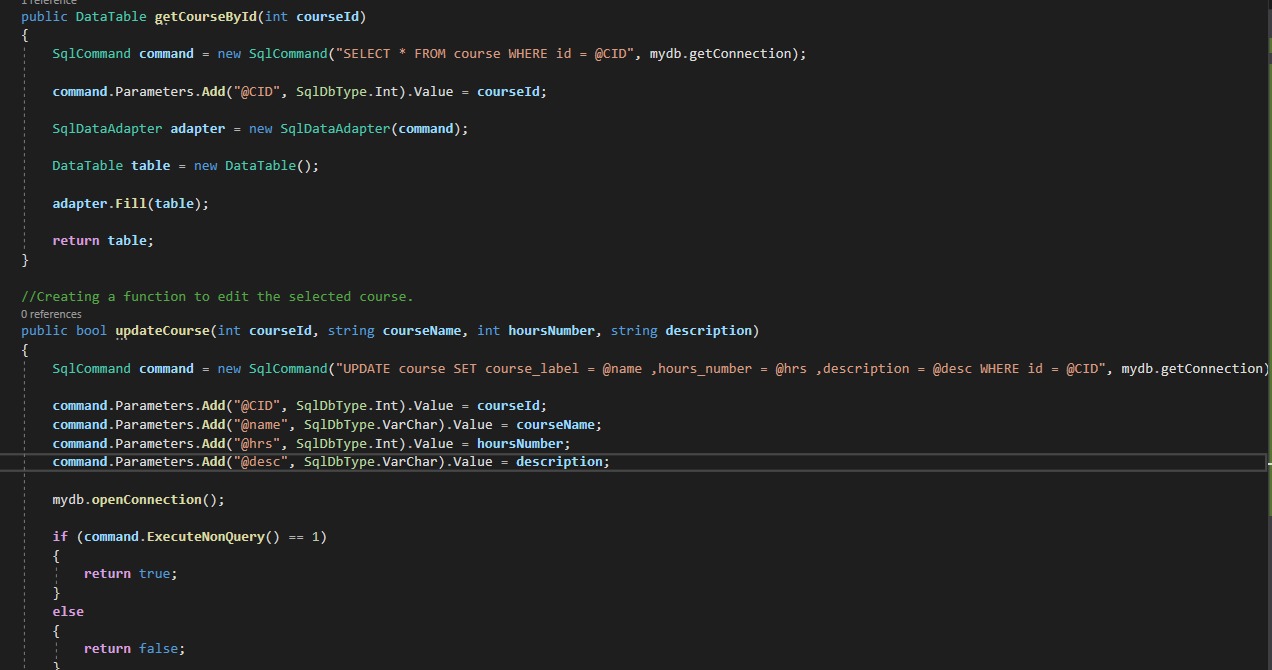
****

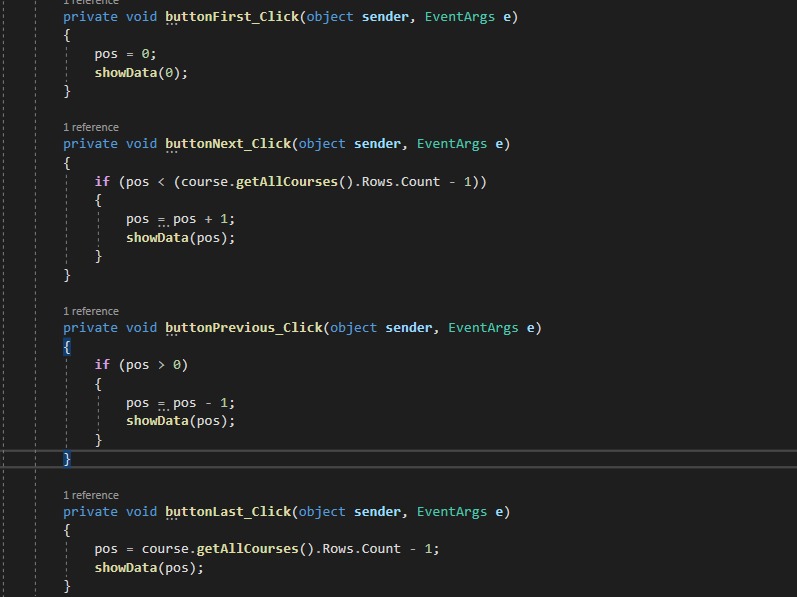
****

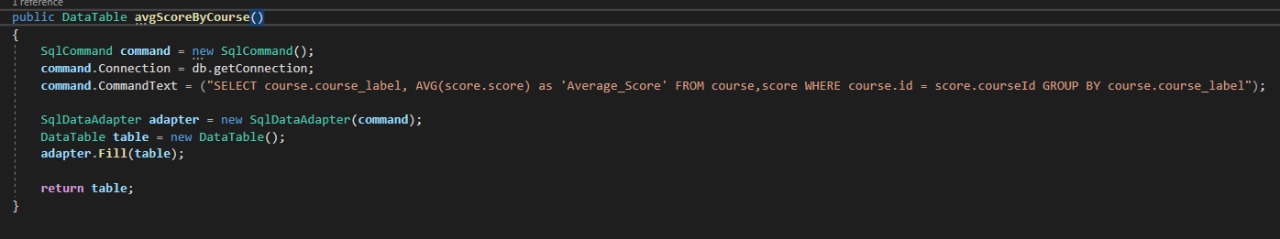
****

****

****

****

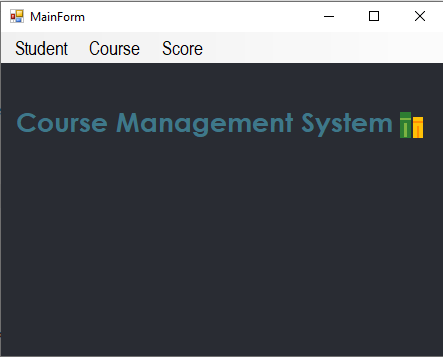
****

****

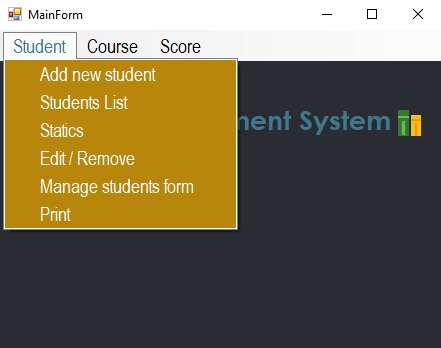
**GUI Specification:**

The Graphical user interface of our student course management system is based on modern GUI`s with many modern and interactive features. The detail of our GUI is given below.

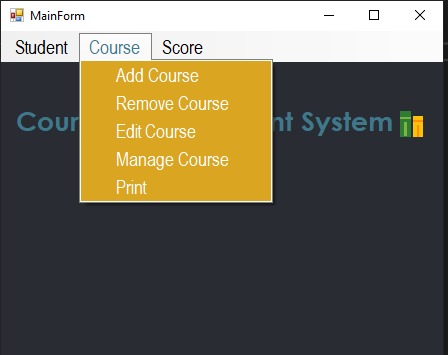
This is the main form, every feature is implemented in this form:



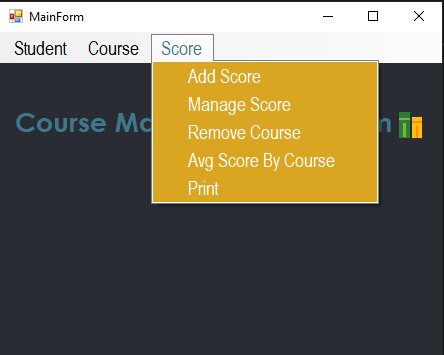
This is student tab with all forms made for students:

****

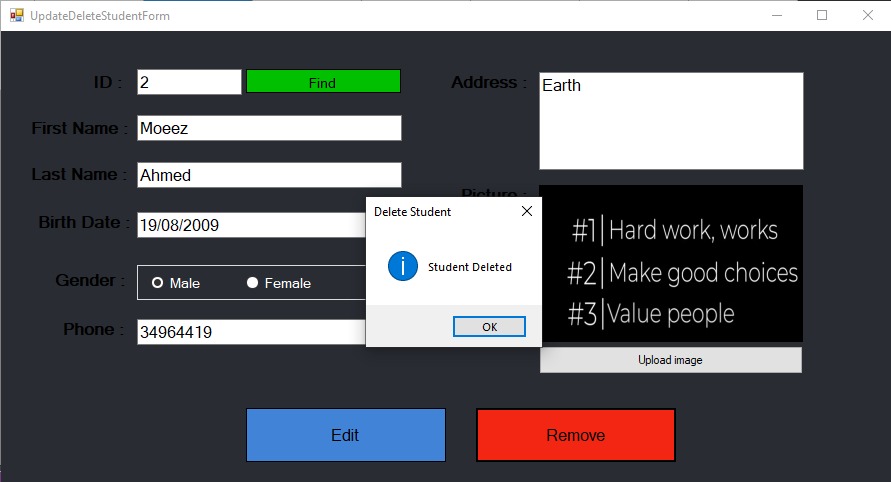
This is Course tab with all forms made for Course Management:

****

This is Score tab made for functionalities of score of student in each course:

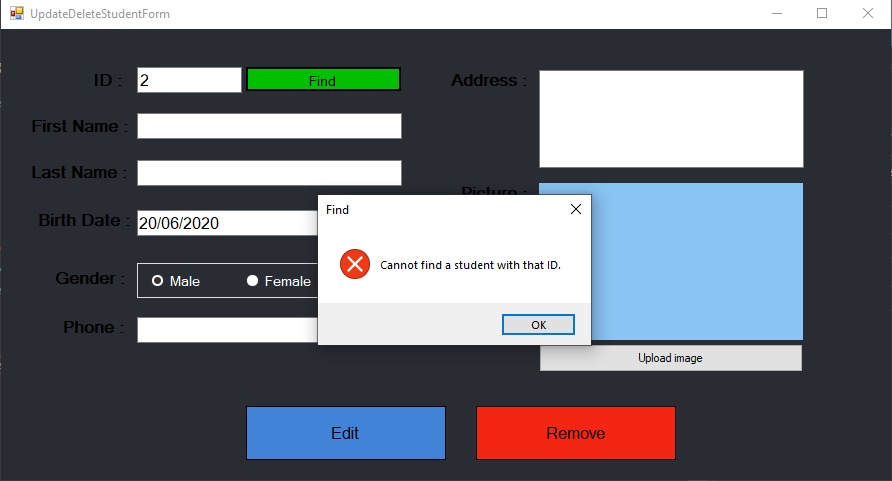
****

This is update/Delete student form:

****

Functionalities include uploading image for updating and all relevant fields. Information of a certain student can also be found by typing student ID and pressing **Find**.

All kind of exceptions are also handled.

****

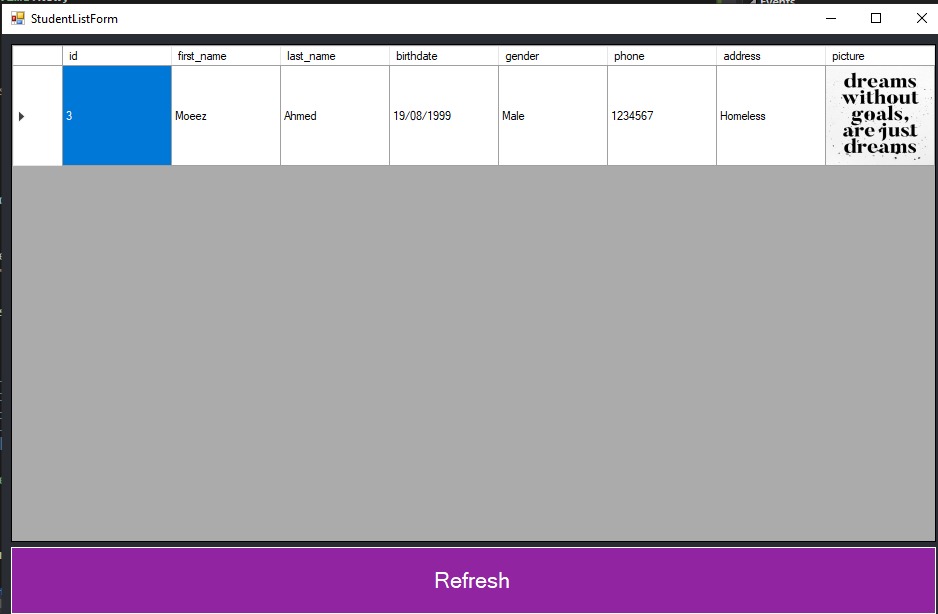
This is add student form:

****

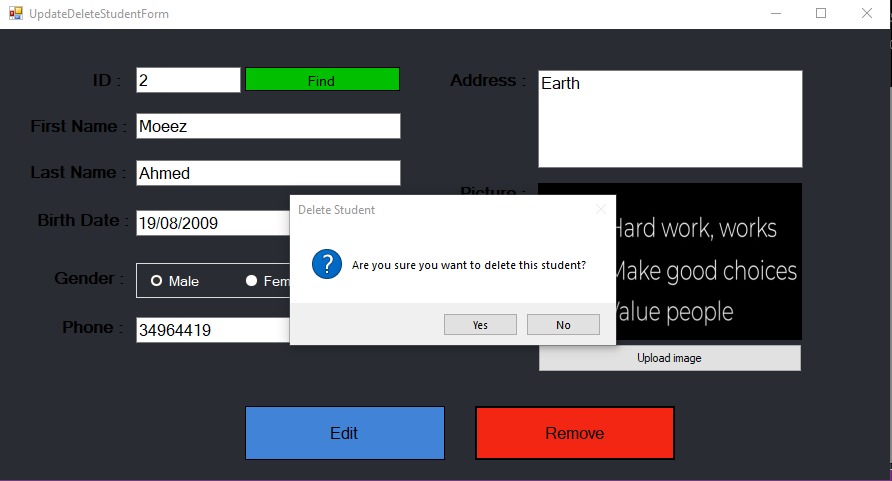
The cancel will just close this form to head back to the main form.

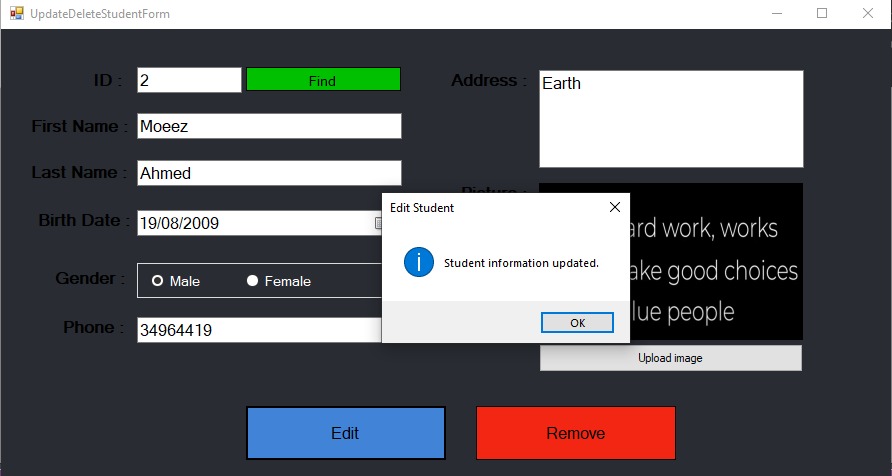
This is student List form:

Information about each student present in database is showed here for further management.

****

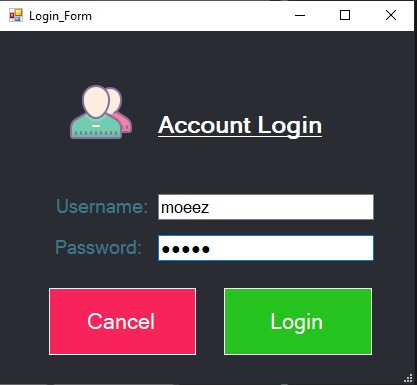
**Student update Delete form:**

****

****

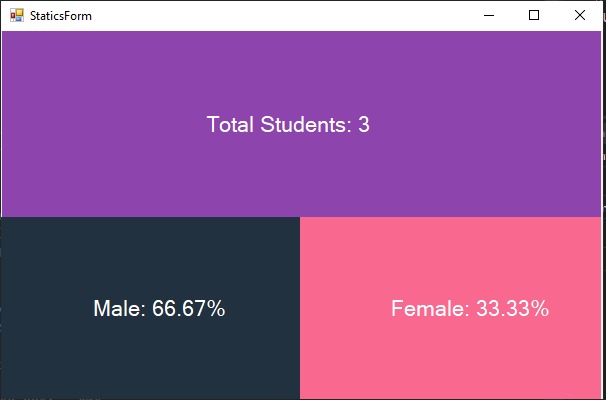
This is login form:

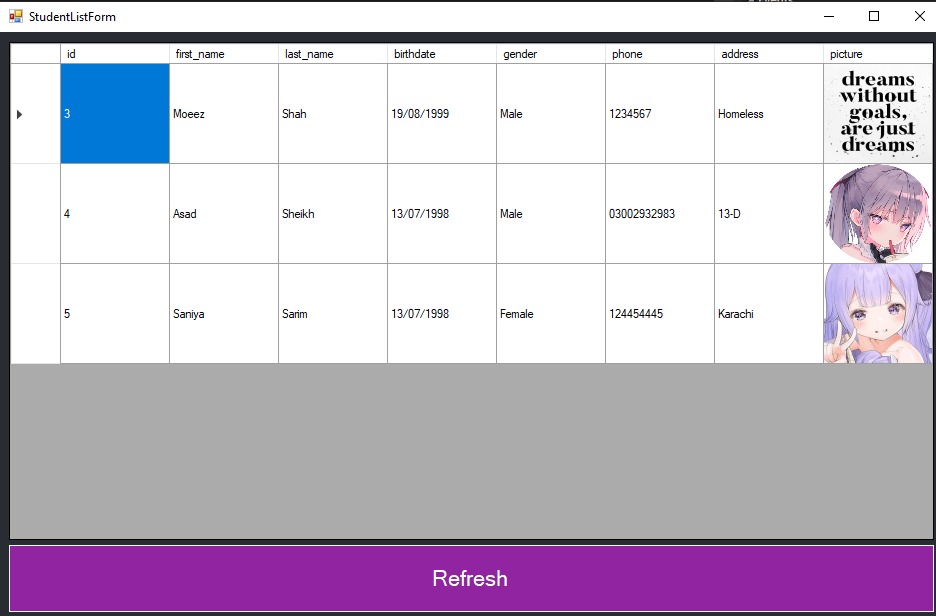
Login information is authenticated from the database in order to continue.

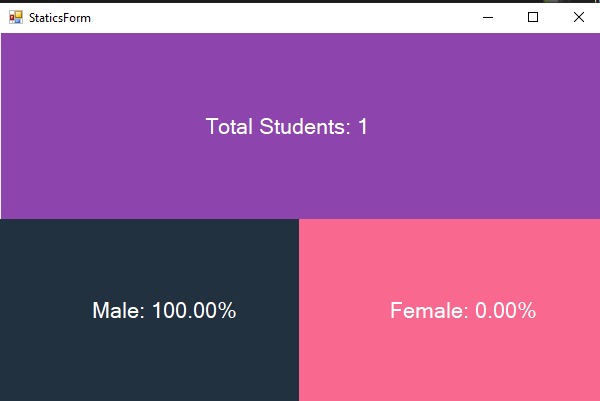
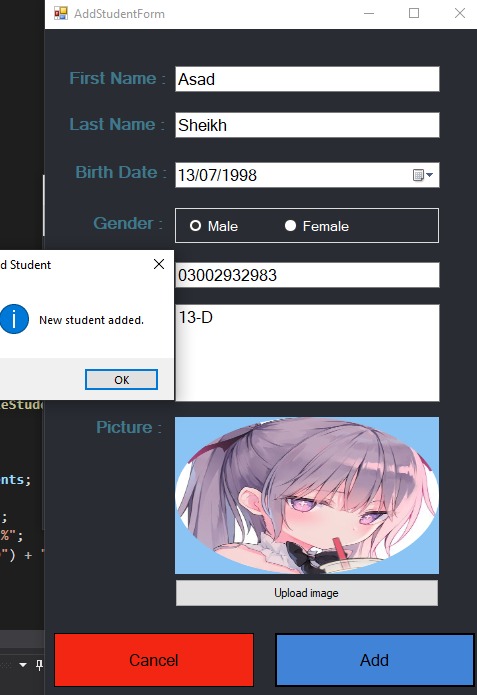
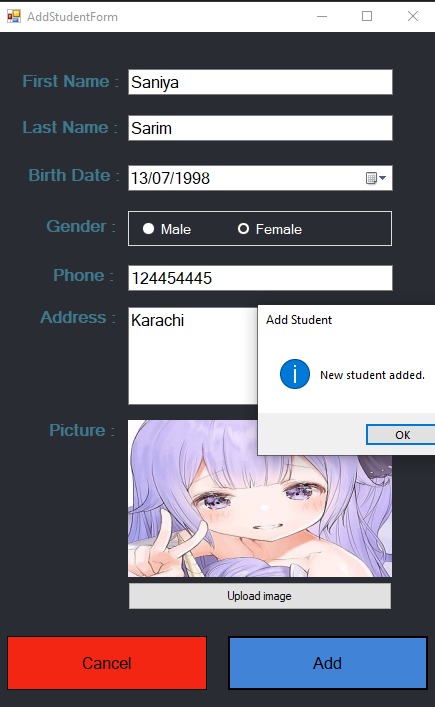
****

This is statics student form:

The total student and the % of male and female are shown in this form.

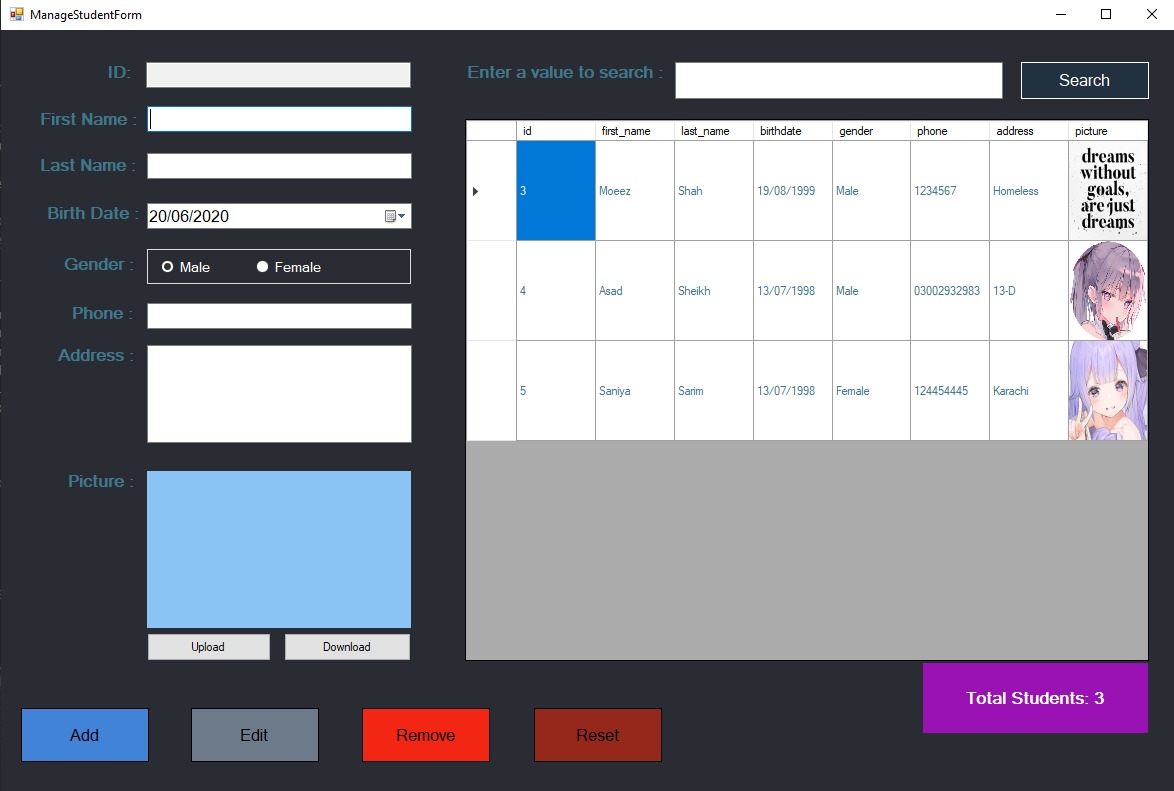
****

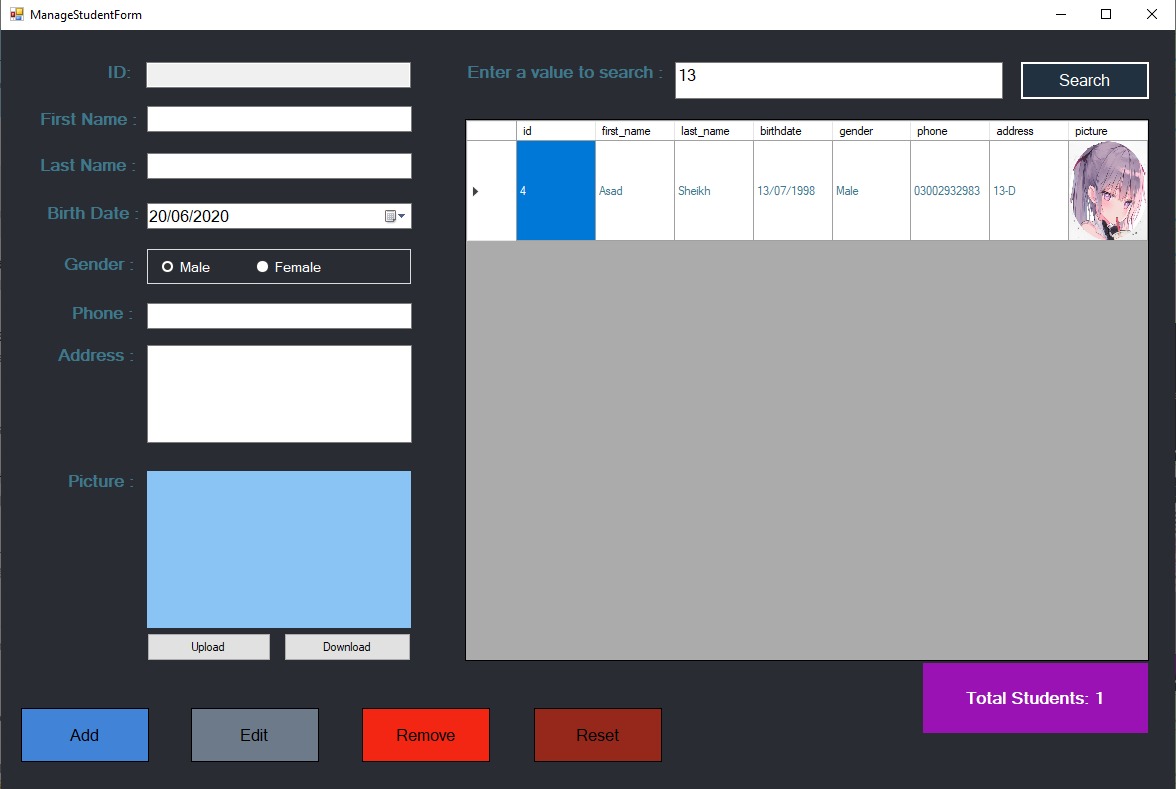
****

****

This is manage student form:

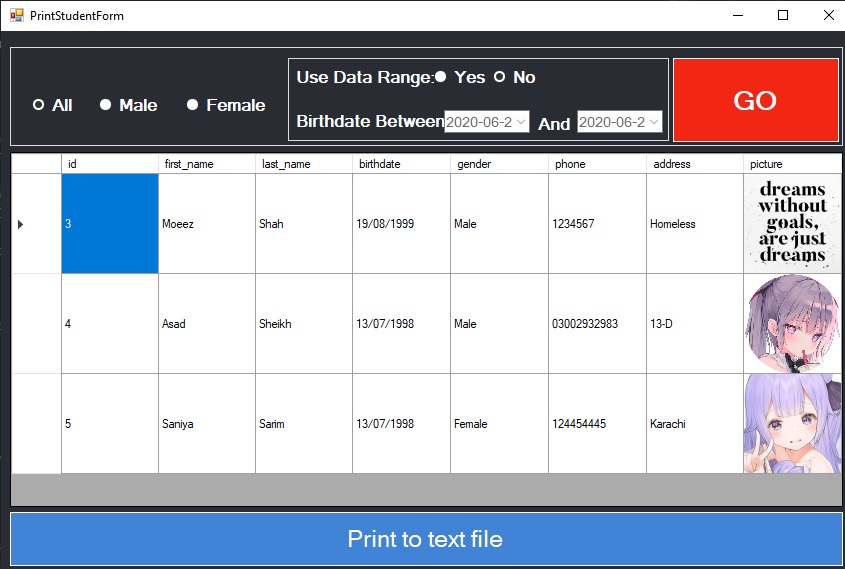
This form is made to perform almost all actions and by using search a specific word can be searched to get further information of a student in data grid view.

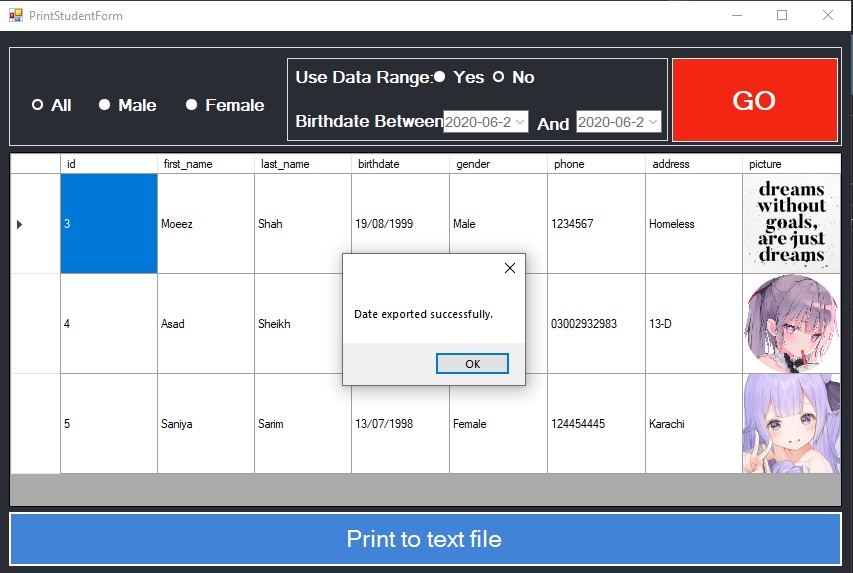
****

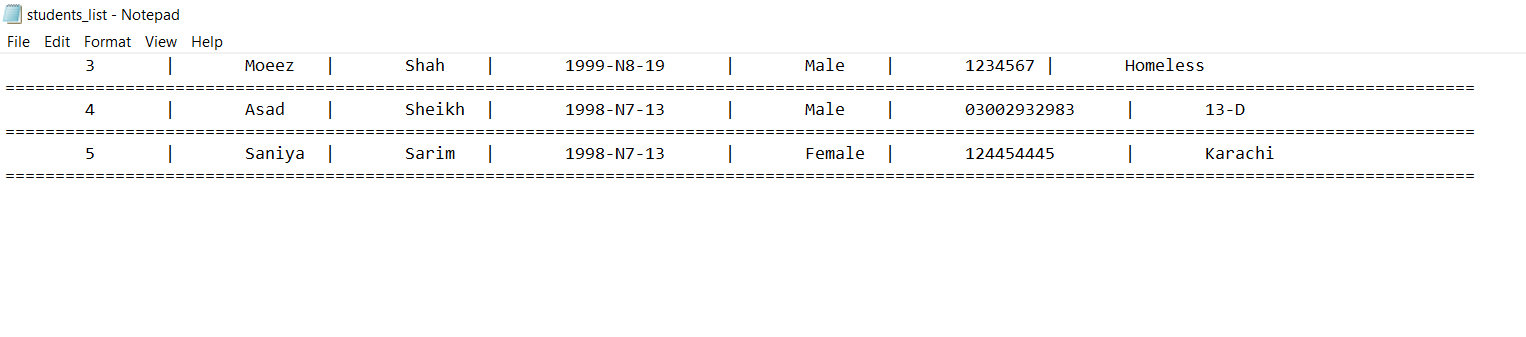
****

This is print student form:

The information displayed will be printed to text file. Further more we can search information based on there gender or if their birthdate exists between certain years.

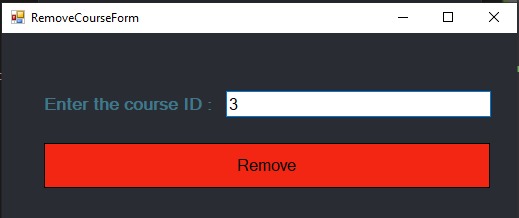
****

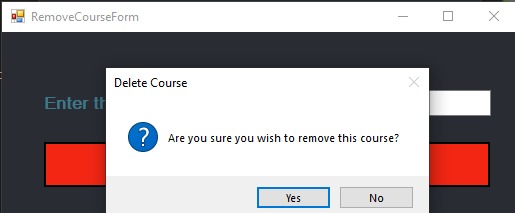
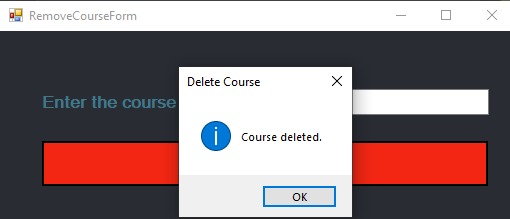
****



This is Course remove form:

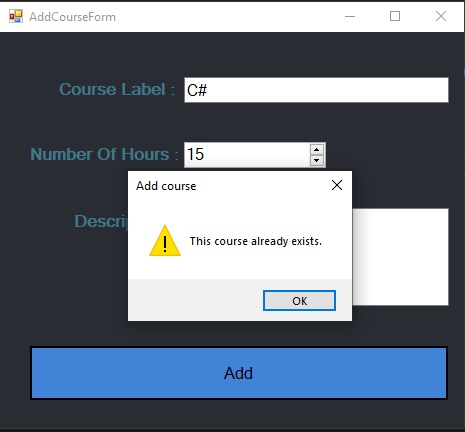
The course can be removed using it`s ID.

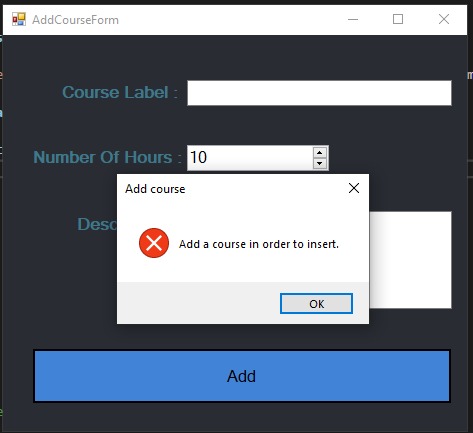
****

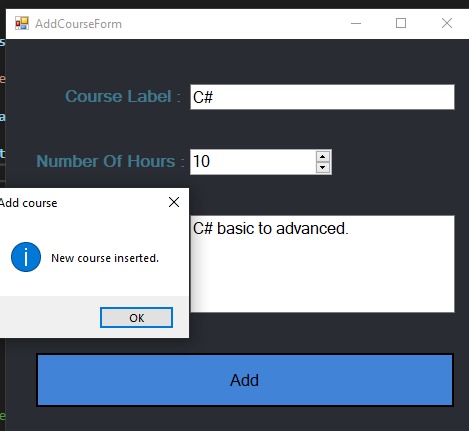
****

This is add course form:

Same course cannot be added twice. Adds the course in database.

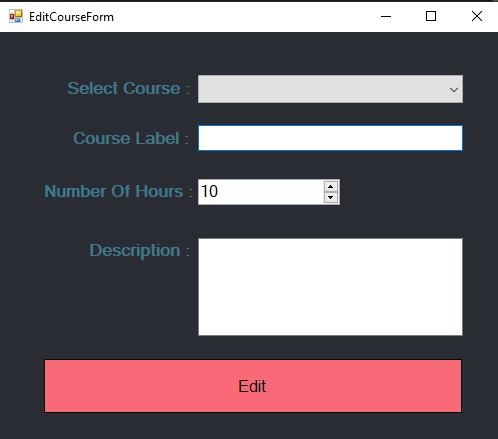
****

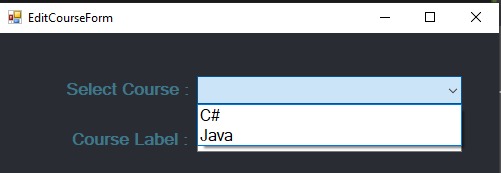
****

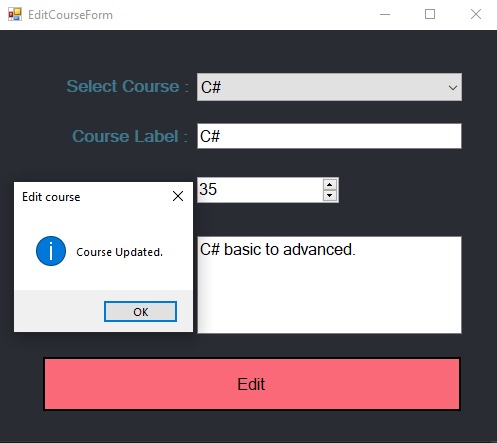
****

This is edit course form:

The course details can be updated in database using this form.

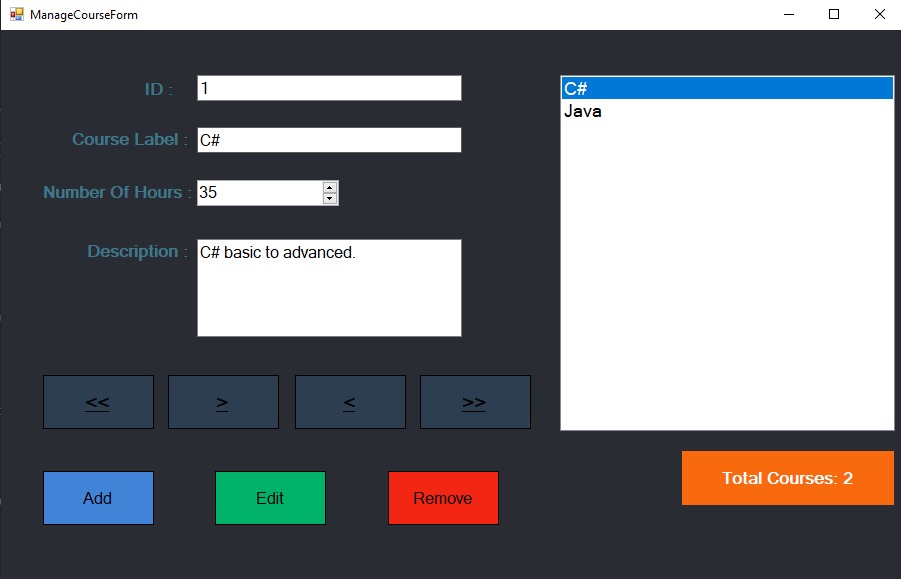
****

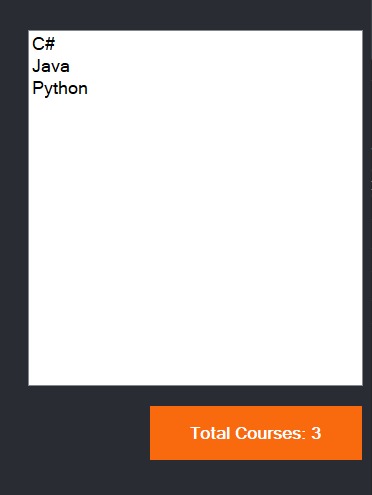
****

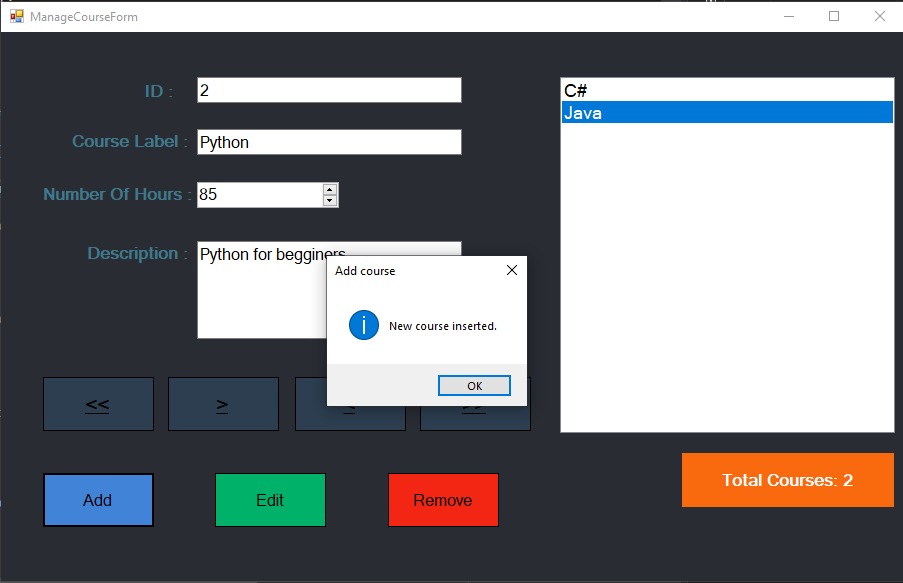
****

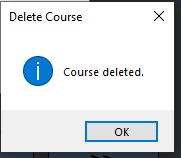
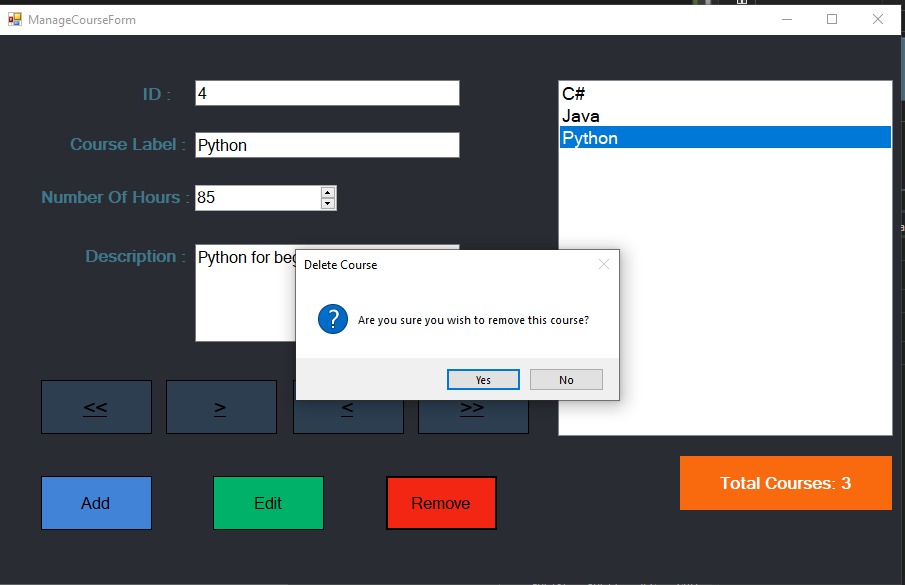
This is manage course form:

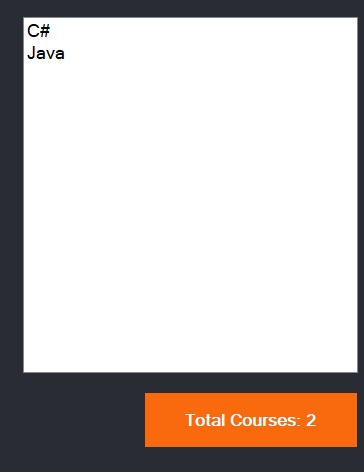
Its basically for total management of courses, Further more Previous, next, first, last functionality is also implemented for moving effectively in Data grid view.

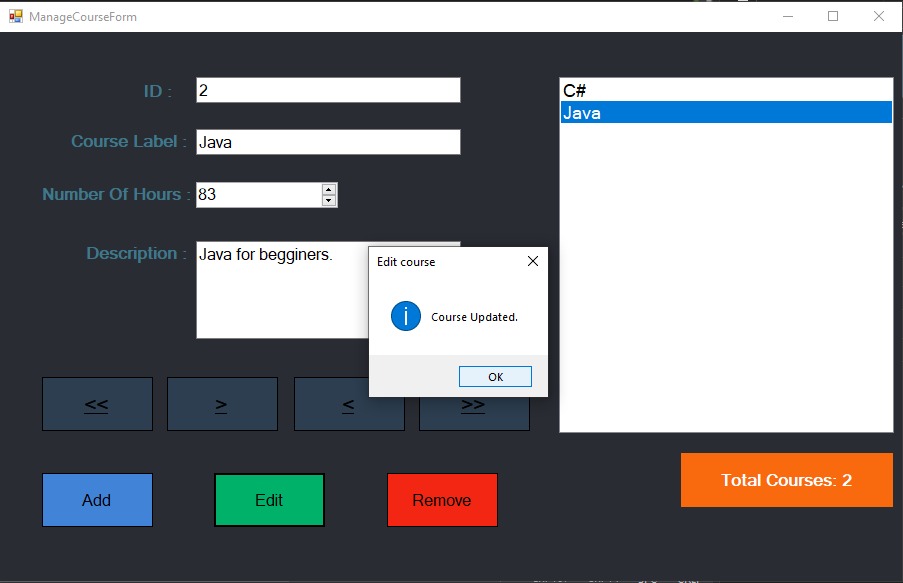
****

****

****

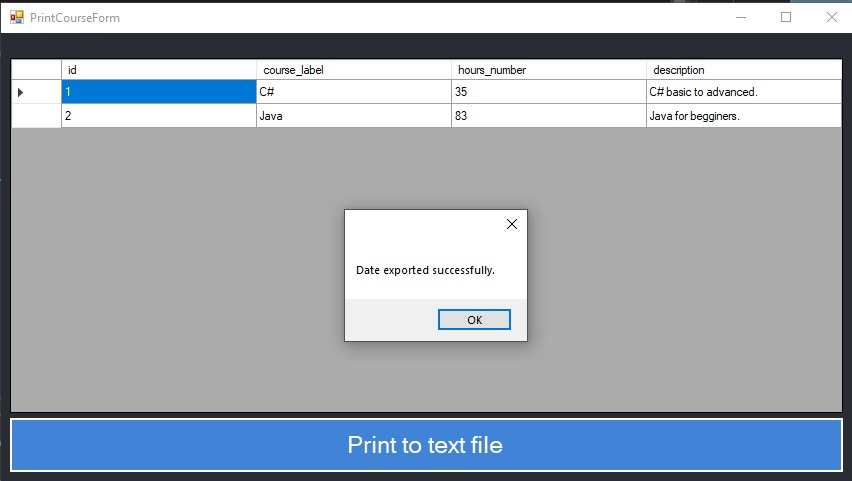
****

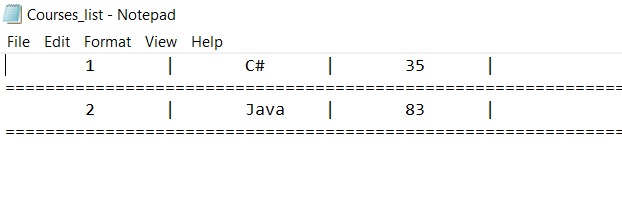
****

****

**This is print course form:**

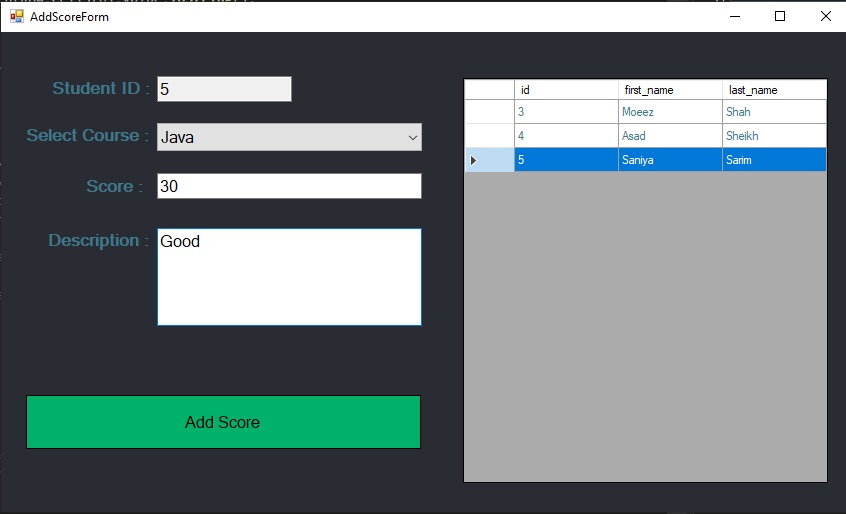
The information here can be printed to a text file.

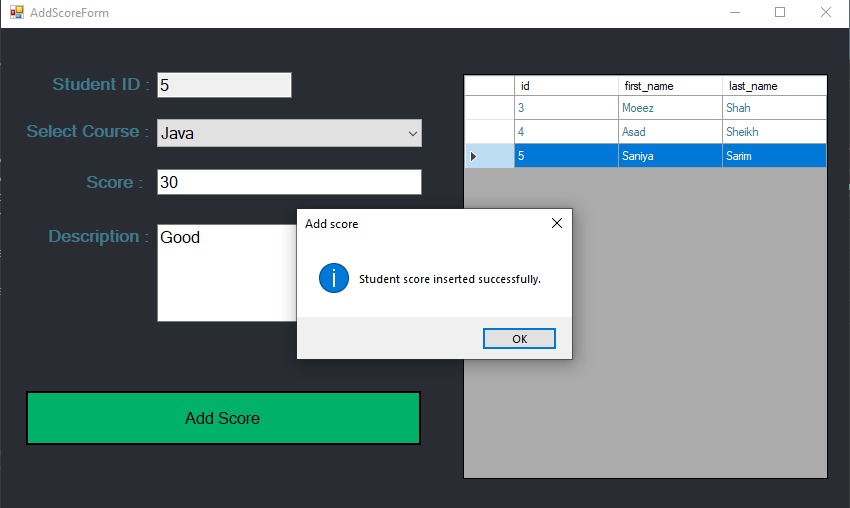
****

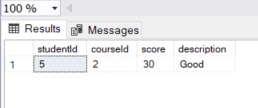
****

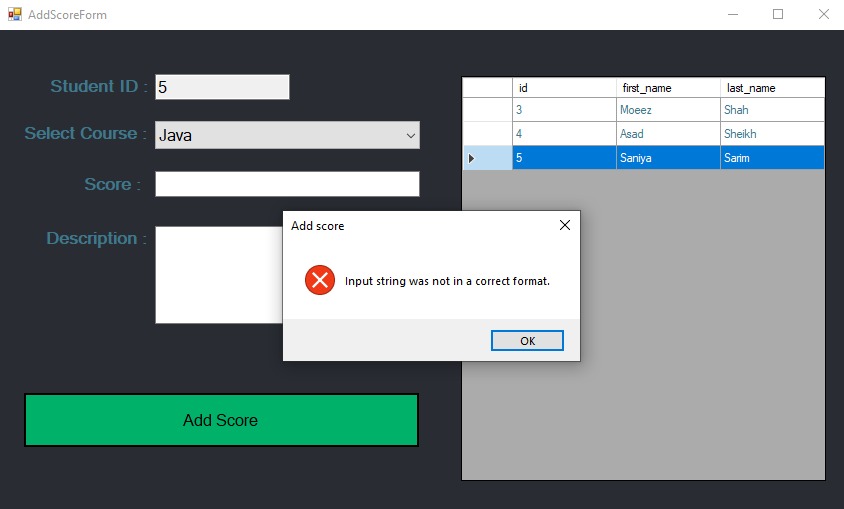
This is add Score form:

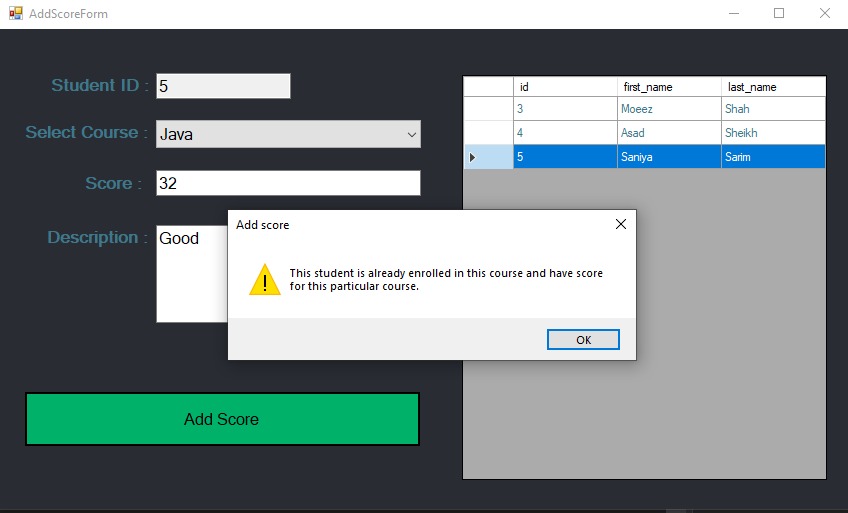
The information of score can be added in this form. Further more the Data grid view is interactive, it automatically fills the needed fields when you press them.

****

****

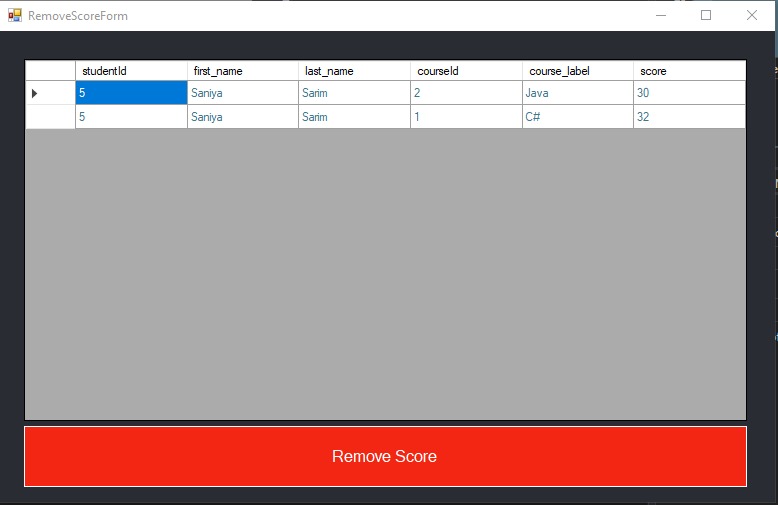
****

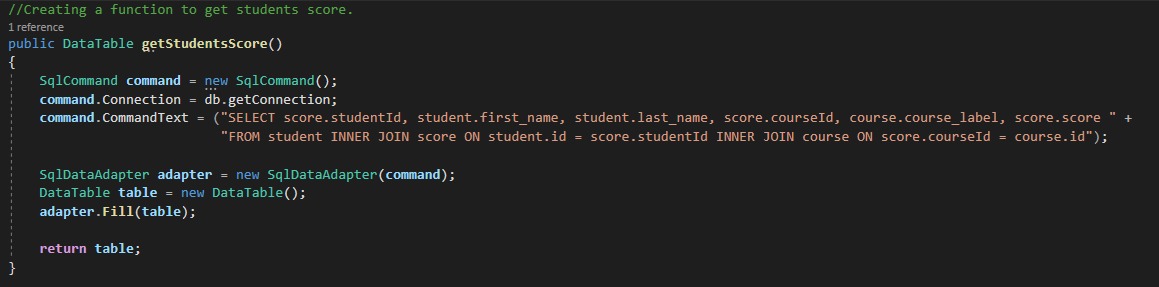
****

****

This is remove score form:

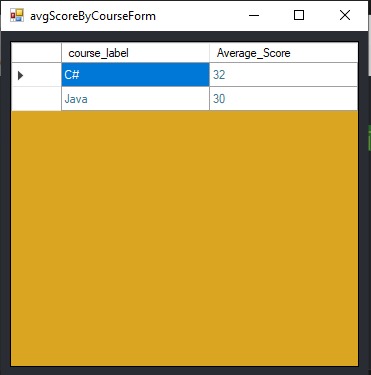
The selected field can be removed from database using this form.

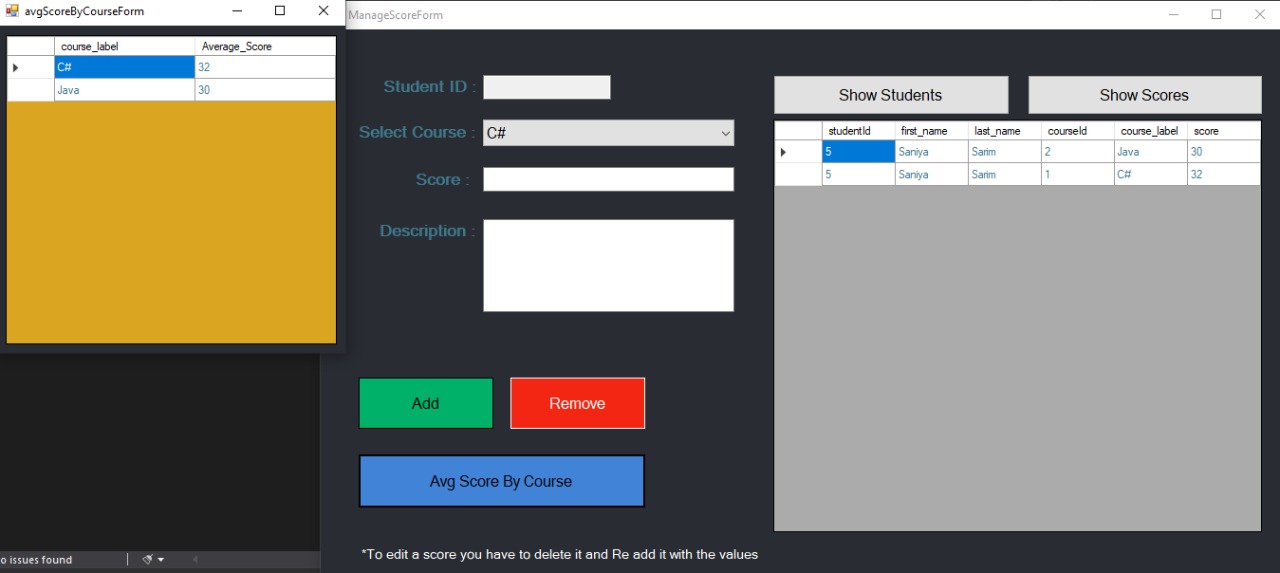
****

****

This is average score by course form:

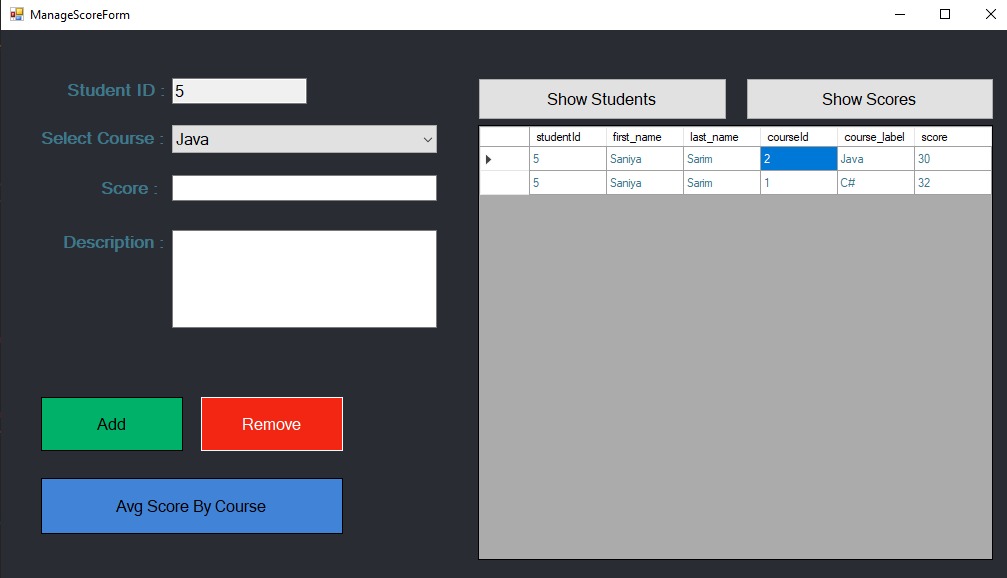
This shows average score of all students in a particular course. (Course average)

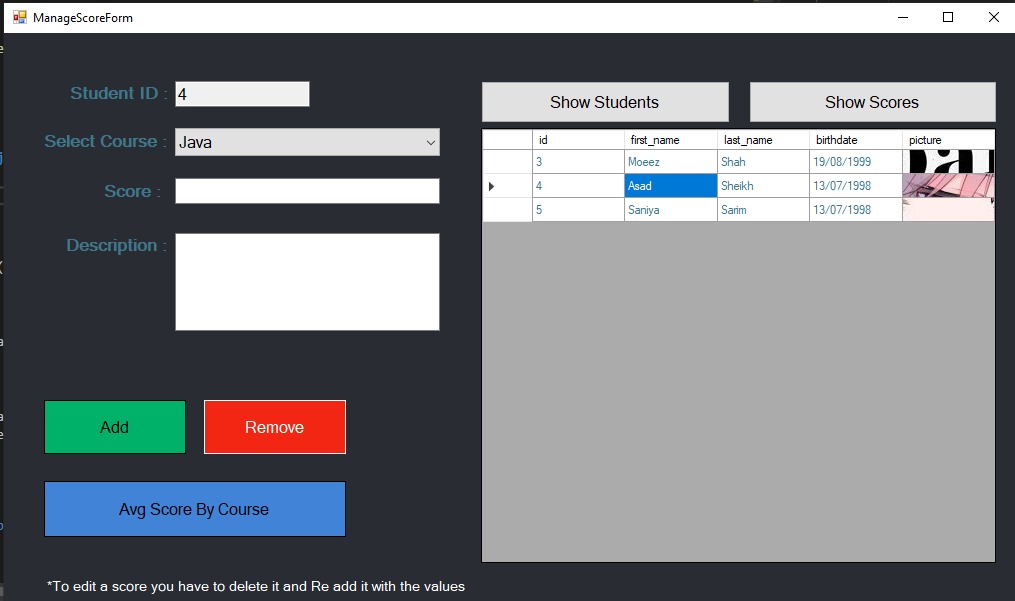
****

****

This is manage Score form:

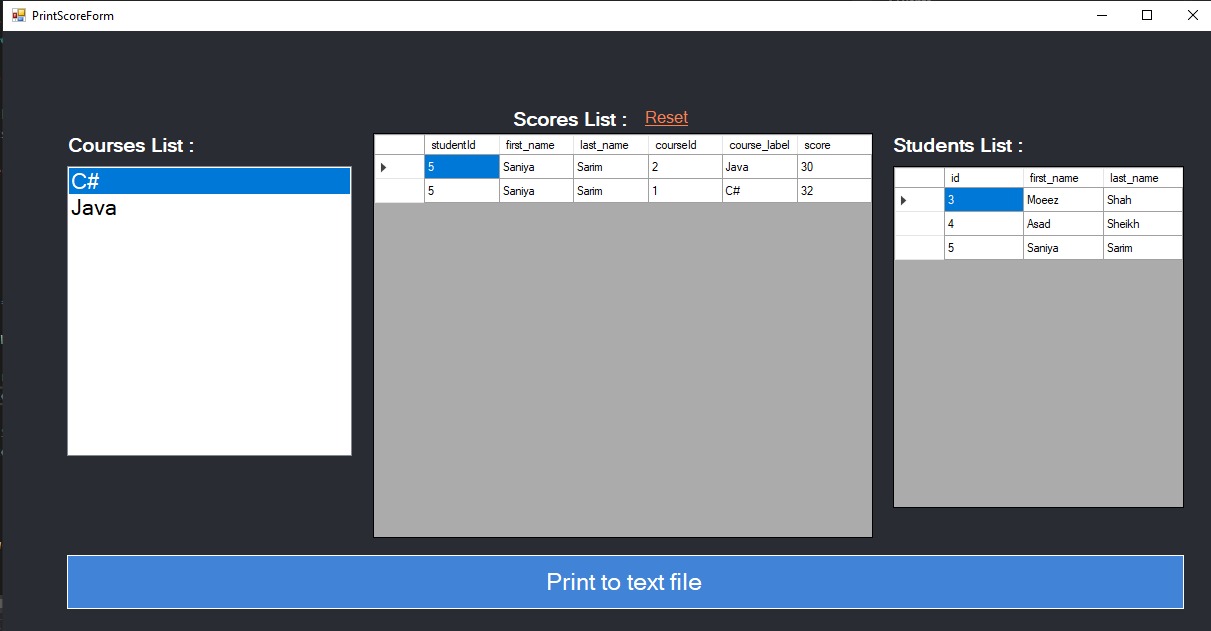
Almost every functionality of Score is present in here. Show students and show scores shows the needed fields, the Data grid view is interactive to user click also.

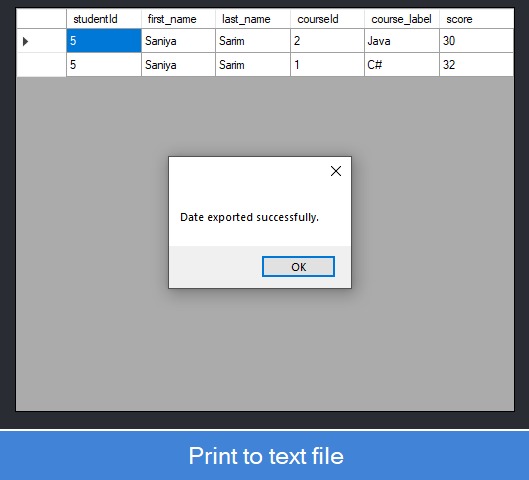
****

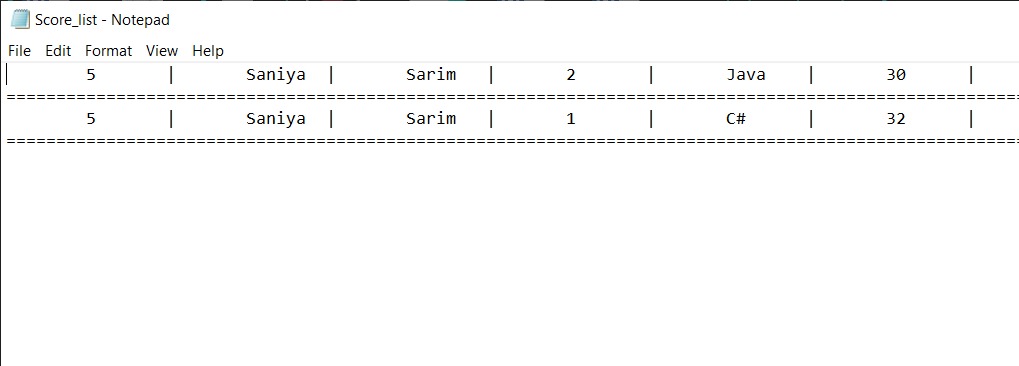
****

This is print score form:

Every List here is interactive to click and changes scores list according to that. The Scores list can be printed to file using this form.

****

****

****

**Conclusion:**

After significant observation and study, a formalized installation procedure for the Student Course Management System was created. Utilizing agile software techniques and common practice human-computer-interaction procedures for testing, our project team accomplished its goal of establishing a usable, scalable, and compatible solution.

The installation package provides for repeatable, understandable, and complete installation routines. Testing indicates that educators of average computer literacy can in fact install the Student Course Management System toolset easily. Now, educators can continue using Student Course Management System as they begin teaching new courses. This results in higher educator productivity and satisfaction with the system.

**References:**

<https://www.w3schools.com/>

<https://www.geeksforgeeks.org/>

<https://docs.microsoft.com/en-us/dotnet/csharp/>