Table of Contents

[1. The common language specification contribution towards the language interoperability 2](#_Toc38477968)

[1.1 Common Type system & Common Language Specification 2](#_Toc38477969)

[1.1.1 Common Type System 2](#_Toc38477970)

[1.1.2 Common Language Specification 3](#_Toc38477971)

[2.Comparetion of selected development tools and technologies for the above mention application 4](#_Toc38477972)

[2.1 The .NET framework 4](#_Toc38477973)

[2.2 C# and Visual Basic 4](#_Toc38477974)

[2.2.1 Features of Visual Basic .NET not found in C# 5](#_Toc38477975)

[2.2.2 Features of C# not found in Visual Basic .NET 7](#_Toc38477976)

[2.3 F# 8](#_Toc38477977)

[2.4 Universal Windows Platform. 8](#_Toc38477978)

[3.System Architecture Diagram 9](#_Toc38477979)

[4.Use case diagram 10](#_Toc38477980)

[5.Data Model Design 12](#_Toc38477981)

[7. Test technique 23](#_Toc38477982)

[7.1 Test Driven Development 24](#_Toc38477983)

[7.2Test cases 25](#_Toc38477984)

[8. Explanation of deployment techniques that are selected and critically evaluation of the suitability of the selection. 41](#_Toc38477985)

[8.1 XCOPY Deployment 41](#_Toc38477986)

[8.2 Copying a Website 42](#_Toc38477987)

[8.3 Creating a Setup Project 42](#_Toc38477988)

# 1. The common language specification contribution towards the language interoperability

## 1.1 Common Type system & Common Language Specification

There are two terms that are freely used in the .NET world, they are very crucial to understand how a .NET implementation enables multi-language development and to understand how it works.

### 1.1.1 Common Type System

To initiate from the beginning, remember that a .NET implementation is language agnostic. This doesn't just mean that a programmer can write their code in any language that can be compiled to IL. It also means that they need to be able to interact with code written in other languages that are can be used on a .NET implementation.

In order to do this transparently, there should be a common way to describe all supported types. This is what the Common Type System (CTS) is in charge of doing. It was designed to do several things:

* Establish a framework for cross-language execution.
* Provide an object-oriented model to support implementing various languages on a .NET implementation.
* Define a set of rules that all languages ought to follow when it comes to working with types.
* Provide a library that contains the basic primitive types that are being used in application development (such as, Boolean, Byte, Char etc.)

CTS defines two major kinds of types that should be supported: reference and value types. Their names point to their definitions.

Reference types' objects are represented by a reference to the object's actual value; a reference here is similar to a pointer in C/C++. It simply refers to a memory location where the objects' values are. Moreover, this has a profound impact on how these types are used. If you assign a reference type to a variable and then pass that variable into a method, for instance, any changes to the object will be reflected on the main object; there is no copying.

Value types are the opposite, where the objects are represented by their values. If you assign a value type to a variable, you are vitally copying a value of the object.

CTS defines several categories of types, each with their specific semantics and usage:

* Classes
* Structures
* Enums
* Interfaces
* Delegates

CTS further defines all other properties of the types, such as access modifiers, what are valid type members, how inheritance and overloading works and so on. Unfortunately, moving deep into any kind of those is beyond the scope of an introductory article such as this, but you can consult [More resources](https://docs.microsoft.com/en-us/dotnet/standard/common-type-system#more-resources) section at the end for links to more in-depth of content that covers these topics.

### 1.1.2 Common Language Specification

To enable full interoperability scenarios, all objects that are created in code must rely on some commonality in the languages that are consuming them (are their callers). Since there are numerous different languages, .NET has specified those commonalities in something which is called the **Common Language Specification** (CLS). CLS defines a set of features that are required by many common applications. It also provides a sort of recipe for any language that is implemented on top of .NET on what it needs to support.

CLS is a subset of the CTS. It means that all the rules in the CTS can apply to the CLS, unless the CLS rules are stricter. If a component is built using only the rules in the CLS, that is, it exposes only the CLS features in its API, it is said to be **CLS-compliant**. For instance, the <framework-libraries> are CLS-compliant precisely because they need to work across all of the languages that are supported on .NET.

You can refer the documents in the [More Resources](https://docs.microsoft.com/en-us/dotnet/standard/common-type-system#more-resources) section below to get an overview of all the features in the CLS.

# 2.Comparetion of selected development tools and technologies for the above mention application

## 2.1 The .NET framework

Over the years, several productivity tools have been developed for specific programming languages and software technologies. The .NET framework is no exception when it comes to have its development tools; these tools will help .NET developers to build and develop a wide range of applications with the use of the framework.

.NET is a free, open-source, and cross-platform framework for building different types of applications. This Microsoft development platform is one of the top platforms for building enterprise software applications for years. With the usage of the .NET framework, software developers can use multiple programming languages, editors, and libraries to build application models for the web, mobile, desktop, gaming, machine learning, artificial intelligence, and the Internet of Things (IoT).

The .NET framework is one of the most productive platforms for software developers because it helps them in developing high-quality applications much faster. The framework can also be used with several programming languages such as C#, F#, Iron Python, Visual Basic, and TypeScript.

## 2.2 C# and Visual Basic

Productive, multi-purpose, type-safe, object-oriented, open source C# and Visual Basic are programming languages designed for creating a variety of applications that run on the .NET Framework. These languages are very powerful, type-safe, and object-oriented.

C# is the specially designed language for Windows application. The C# is the multi-paradigm and the object-oriented programming language developed by Microsoft as a part of a .NET initiative. It is easy to create an application in C# programming language. Because of this language developers can create an excellent and interactive app in Windows. The C# owns combined features of both C and C++ language. Therefore, c# is the best language for this application.

### 2.2.1 Features of Visual Basic .NET not found in C#

* Variables can be declared using the With Events construct. This construct is available so that a programmer may select an object from the Class Name drop down list and then select a method from the Declarations drop down list to have the Method signature automatically inserted.
* Auto-wire up of events. VB.NET has the Handles syntax for events, which connects event handlers to object variables rather than to objects.
* Firing of events is done with the Raise Event keyword, giving the IDE the chance to show a list of available events to pick from. Raise Event implicitly checks if there are any event handlers wired up. (in C# raising an event is syntactically identical to calling a procedure, and it requires an additional line of code to check for wired event handlers)
* Delegates for events don't need to be declared. They are implicitly declared in the declaration of the events.
* Referring to an object using an unqualified dot reference, using the with ... End With structure
* XML Literals
* Inline date declarations, e.g. #12/31/2000#
* Module (although C#'s static classes with additional semantics, but each field must be individually declared as static)
* Members of Modules imported to the current file, can be accessed with no preceding container accessor
* The My namespace.
* COM components and interoperability were more powerful in VB.NET, as the Object type is bound at runtime; however, C# 4.0 added the dynamic type, which functions as a late-bound form of Object
* Namespaces can be imported at the project level, so that they don't have to be imported into each individual file, as in C#
* Definition of conditional compiler constants
* Property methods may take parameters
* Properties can be passed to methods with by ref parameters (ref parameters in C#). In C# you need to write three additional instructions: Declare a variable, copy the property value into the variable and copy the variable back to the property after the method call.
* Enums can be defined inside interfaces
* Case statements may contain inequality expressions, like Is >= 3. (in C# this can be mimicked by nested Else and If statements)
* Overloads keyword specifies that a property or procedure redeclares one or more existing properties or procedures with the same name within the same class or the base class. (the lack of this keyword in C# might lead to inadvertent overloading)
* Implements keyword to indicate which interfaces a class member implements. In C# a similar syntax exists, but it is optional and it can only be applied if the member implements a single interface.
* Like operator for pattern comparison of strings in a much simpler way than using regular expressions. (in C# the same is available with the Microsoft visual basic compiler services like operator like string method, but not as a handy language key word)
* Return statement is not required. Return can also be done by assigning the value to the function
* Visual basic has built in constants like vb Crl+f and vb Tab
* No out-parameter modifier exists, because in VB all variables are automatically initialized.
* The My Class keyword behaves like an object variable referring to the current instance of a class as originally implemented. My Class is similar to Me, but all method calls on it are treated as if the method were Not Overridable.
* My Base. New is used to explicitly call a base class constructor from a derived class constructor.
* The My feature provides easy and intuitive access to a number of .NET Framework classes, enabling the Visual Basic user to interact with the computer, application, settings, resources, and so on.
* Local variables (i.e. variables declared inside of a procedure) are automatically initialized.
* Local variables can be declared with the Static modifier in order to preserve their value between calls to the procedure.
* The Default declaration makes a property an index and able to use the shorter syntax for collection retrievals like My Collection C# has a similar construct but it can only declare a single default indexer. In VB one could, for instance, have two indexers my Collection or my Collection("Bob") on a collection with Integer keys and String values.
* C# lacks the Direct Cast (mapping to a single CLR instruction), strict type conversion can be achieved by the as operator which includes an additional runtime error protection.
* C# lacks the End statement which abruptly terminates an application.
* Lambda expressions inference Dim x = Function (n As Integer) n + 1

### 2.2.2 Features of C# not found in Visual Basic .NET

Multi-line comments. In VB this is handled in the Visual Studio IDE editor, which adds comment markers to selections.

Static classes (classes which cannot contain any non-static members, although VB.NET's Modules are essentially static classes with additional semantics)

Can use checked and unchecked contexts for fine-grained control of overflow/underflow checking

Iterative for-loops can contain multiple conditionals, such as for (int i = 0; i < 10 && something True; i++). This is a legacy of C, where the for statement is basically syntactic sugar for a while statement.

The getter and setter of a property may implement separate interfaces. In VB you'd have to define two properties instead: a read-only property implementing one interface, and a write-only property implementing the other interface.

Implicit interface implementation

Can use the coalesce operator ?? to return the first non-null value (ex. null ?? null?? 1 returns 1). VB.NET would have to use the If function with two parameters.

Pointers (in the unsafe context)

Conditional operator? (some Thing True Or False) ? when True() : when False();

Other characteristics of Visual Basic .NET not applicable to C#

Conversion of Boolean value True to Integer may yield -1 or 1 depending on the conversion used

Assigning and comparing variables uses the same token: =. Whereas C# has separate tokens, == for comparison and = to assign a value

VB.NET identifiers are not case-sensitive.

When assigning a value to a variable with a different data type (and with Option Strict not turned on), VB.NET will coerce the value if possible. This automatic coercion sometimes can lead to unexpected results, for example:

## 2.3 F#

Simple, efficient, data-rich, functional-first programming F# is an open source, cross-platform, functional-first programming language that enables developers to tackle computing problems with simple, maintainable and robust code. F# is a general purpose, strongly typed, multi-paradigm programming language that encompasses functional, imperative, and object-oriented programming methods. F# is used to a great extent as a cross-platform Common Language Infrastructure language, but it can also generate JavaScript and graphics processing unit code.

## 2.4 Universal Windows Platform.

One Windows platform, many devices Leverage your skills in C# and Visual Basic to take advantage of the power of Windows 10 to build the next generation of apps. Windows 10 runs on a single, unified core across devices mobile, desktop, Xbox, or elsewhere. Windows 10 apps you build with C# and Visual Basic run as fast as C++ with the .NET Native runtime.

# 3.System Architecture Diagram

Application

Read Service

Command Service

Read Database

Write Database

Send Command

Send Changed Event

Update Read Model

Store Data

# 4.Use case diagram

Student Coordinator

Student

Lecture

# 5.Data Model Design

User name

Password

Login

Result Sheet

Update Profile

Student Dashboard

Add new student

Find student

Student Record

Add user

Student Coordinator Dashboard

Add New Result

Update Results

Result Record

Lecture Dashboard

Add new student

Find student

Student Record

Add user

Student Coordinator Dashboard

Student Record

Enroll number

Full name

National id number

Address

Email

Mobile number

Parent mobile number

Gender

District

User name

A/L Stream

Batch

Course

Birthday

Add new student/ Find student

Result Sheet

Update Profile

Student Dashboard

Enroll number

Full name

National id number

Address

Email

Mobile number

Parent mobile number

Gender

District

User name

Birthday

Update profile

Record Record

Enroll number

Subject

Obtain marks

Full marks

Batch

Add new result/Update result

Add New Result

Update Results

Result Record

Lecture Dashboard

**6. Wireframes**

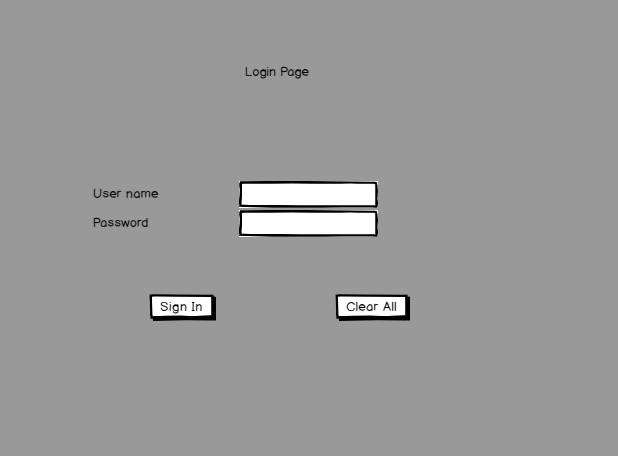


Figure 1 login wireframe

Can logging to particular dashboard according to user type.

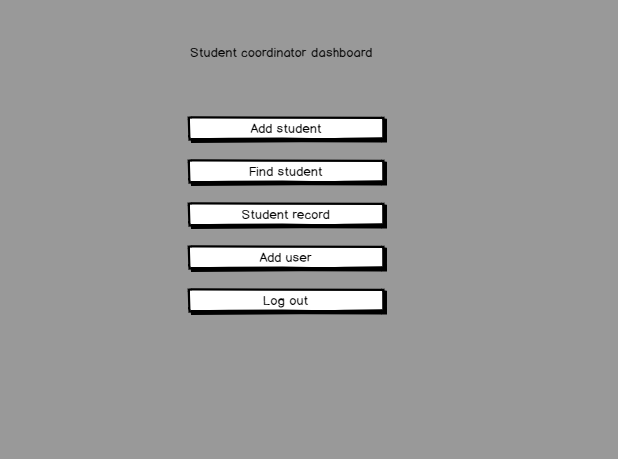


Figure 2 Student Coordinator Dashboard

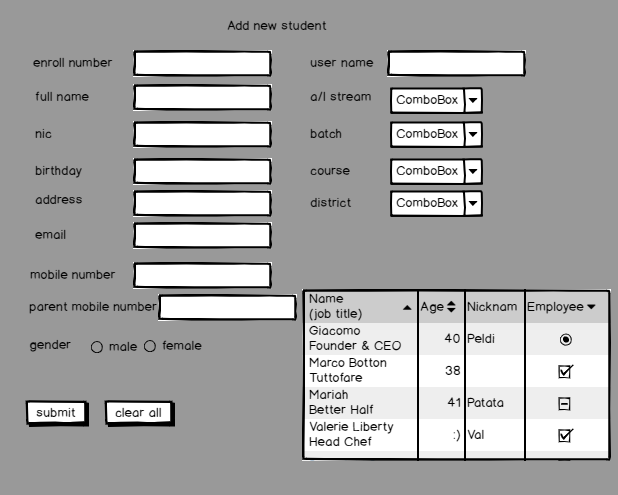


Figure 3 Add New Student wireframe

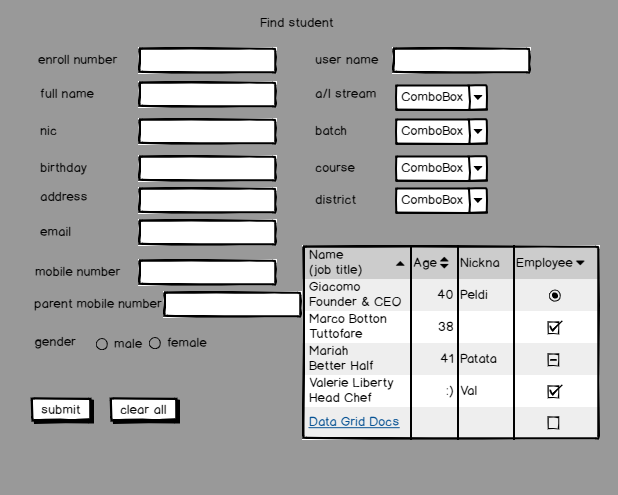


Figure 4 Find Student wireframe

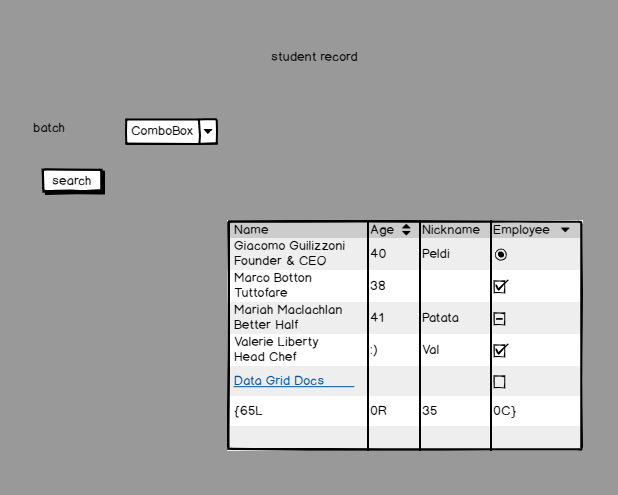


Figure 5 Student Record wireframe

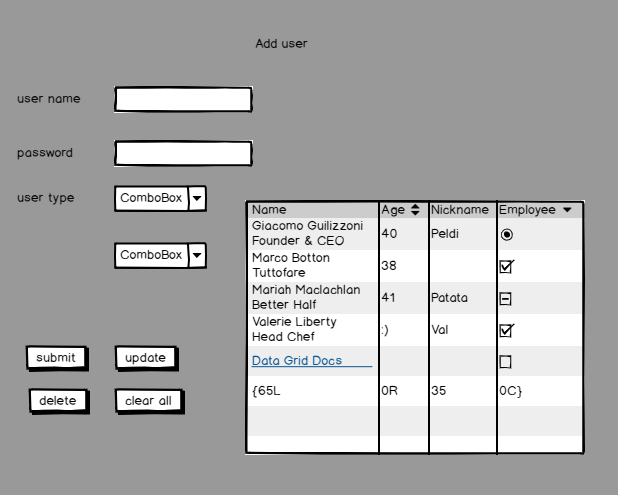


Figure 6 Add user wireframe

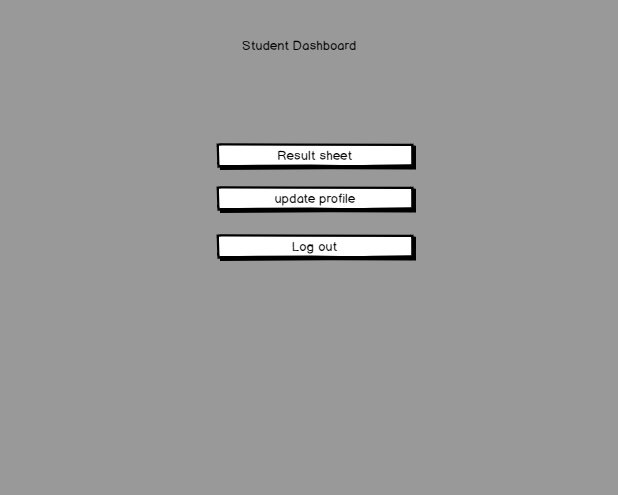


Figure 7 Student dashboard wireframe

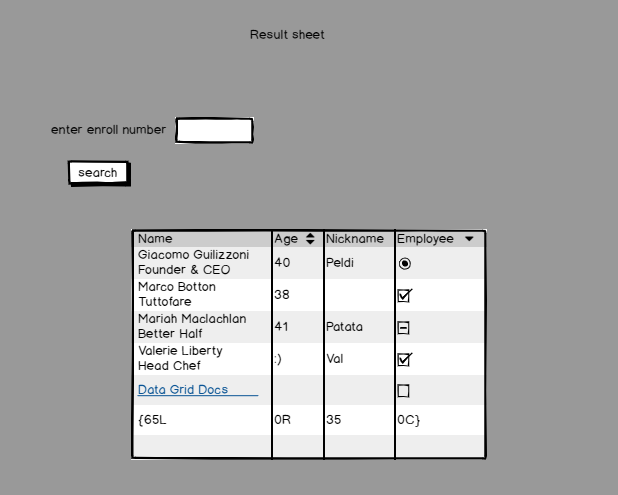


Figure 8 Result sheet wireframe

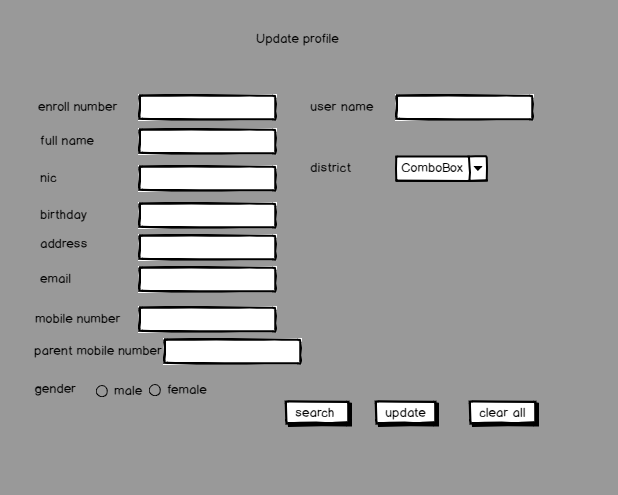


Figure 9 Update profile wireframe

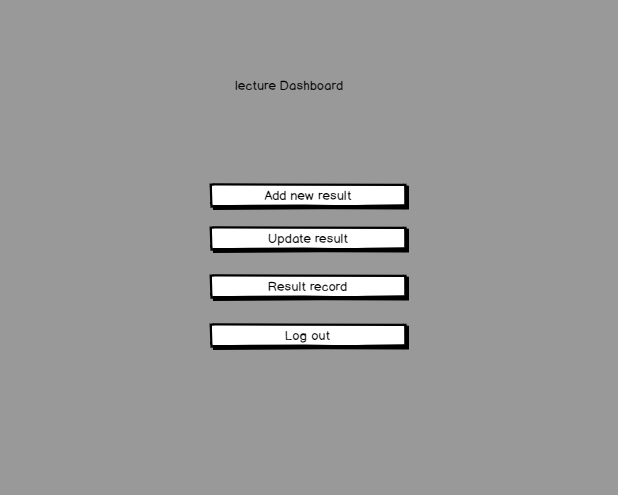


Figure 10 lecture dashboard wireframe

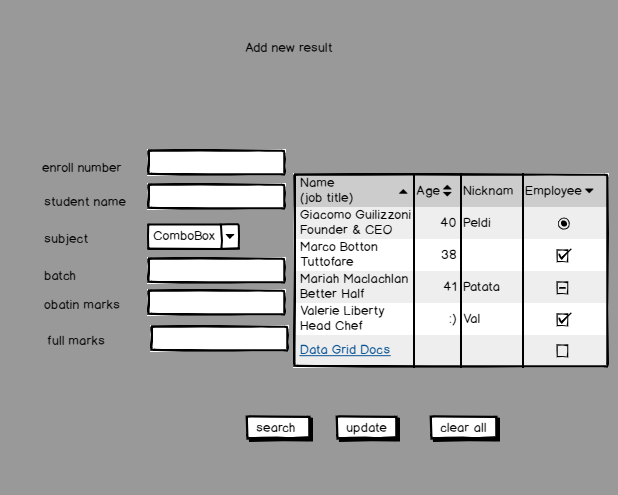


Figure 11 Add new result wireframe

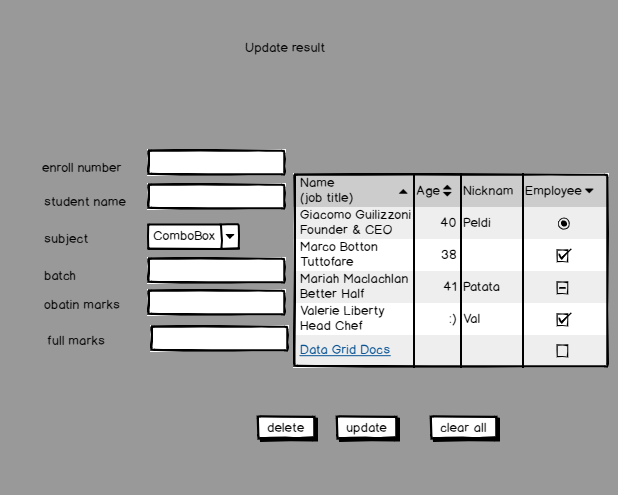


Figure 12 Update Results wireframe

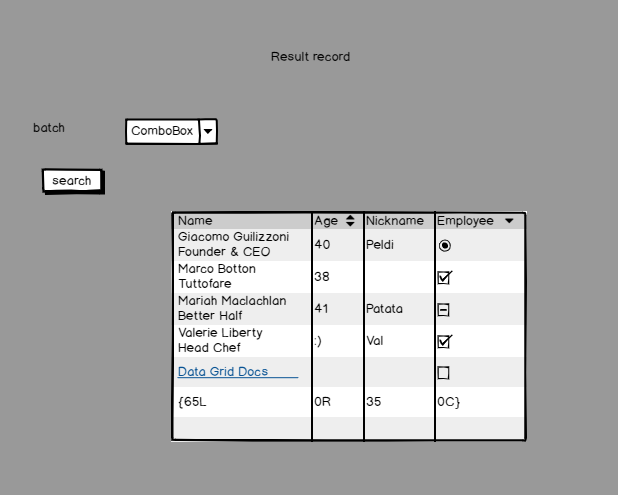


Figure 13 Result Record wireframe

# 7. Test technique

In the software development process Unit Tests basically test individual parts (also called as Unit) of code (mostly methods) and make it work as expected by the programmer. A Unit Test is a code written by any programmer which tests small pieces of functionality of big programs. Performing unit tests is always designed to be simple, A "UNIT" in this sense is the smallest component of the large code part that makes sense to test, mainly a method out of many methods of some class. Generally, the tests cases are written in the form of functions that will evaluate and determine whether a returned value after performing Unit Test is equals to the value you were expecting when you wrote the function. The prime objective in unit testing is isolating a unit part of code and validating it’s to correctness and reliable.

One of the most valuable benefits of using Unit Tests for development is that it gives a positive confidence that our code will work as we expected it to work in our development process. Unit Tests always give the certainty that it will lead to a long-term development phase because with the help of unit tests we can easily understand that our foundation code block is totally dependable on it.

There are few reasons that can provide a basic understanding of why a developer needs to design and write out test cases to make sure major requirements of a module are being validate during testing,

* Unit testing can increase confidence and certainty in changing and maintaining code in the development process.
* Unit testing always has the capability to find problems in early stages in the development cycle.
* Codes are more reusable, reliable and clean.
* Development becomes faster.
* Automation is easier.

I would like to indicate that unit tests are quite easy to use. Unit testing is a way or we can say that process with the help of this a programmer can make themselves sure that their written code is fully functional, clean, reliable and will definitely work as the same way it is supposed to work.

One of the fundamental principles of adopting unit testing is to follow a TDD (Test Driven Development) approach.

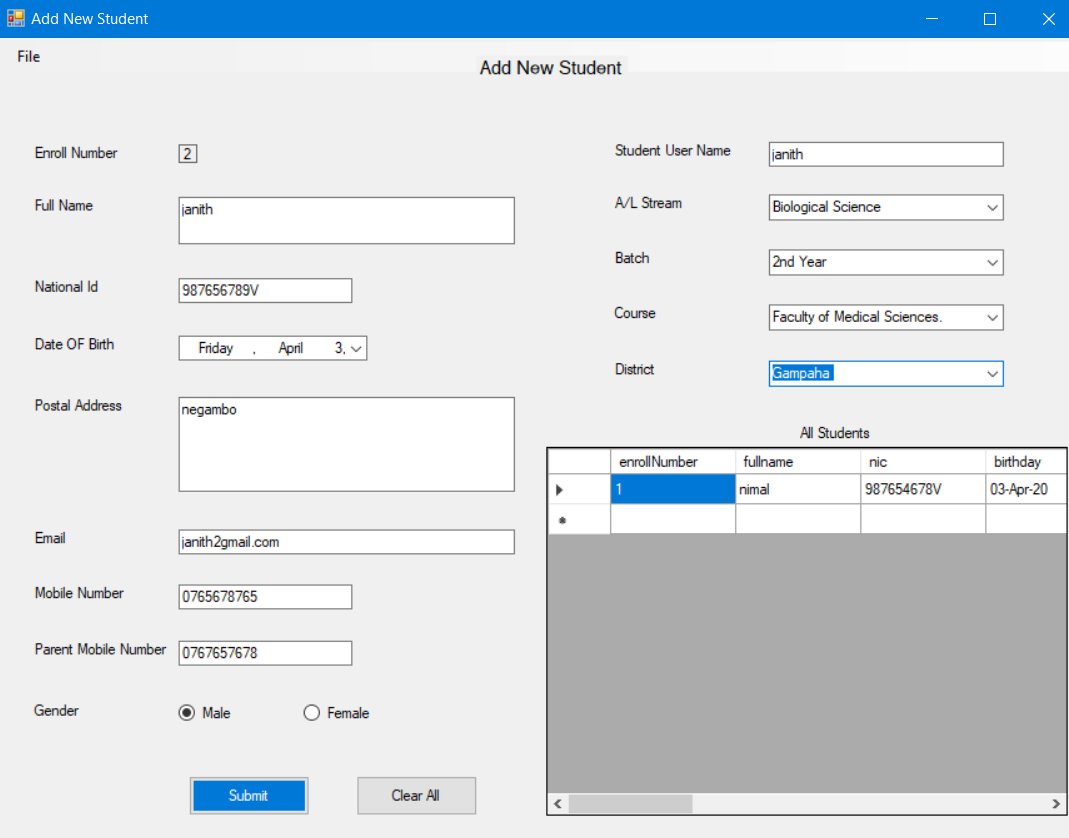
## 7.1 Test Driven Development

Test driven development (TDD) approach first, the test is developed which specifies and validates what the code will do. In simple terms, test cases are created before code is written. The purpose of TDD is to make the code clearer, simple and bug-free.

Test-Driven Development starts with designing and developing tests for every small functionality of an application. TDD instructs developers to write new code only if an automated test has failed. This avoids duplication of code. The full form of TDD is Test-driven development.

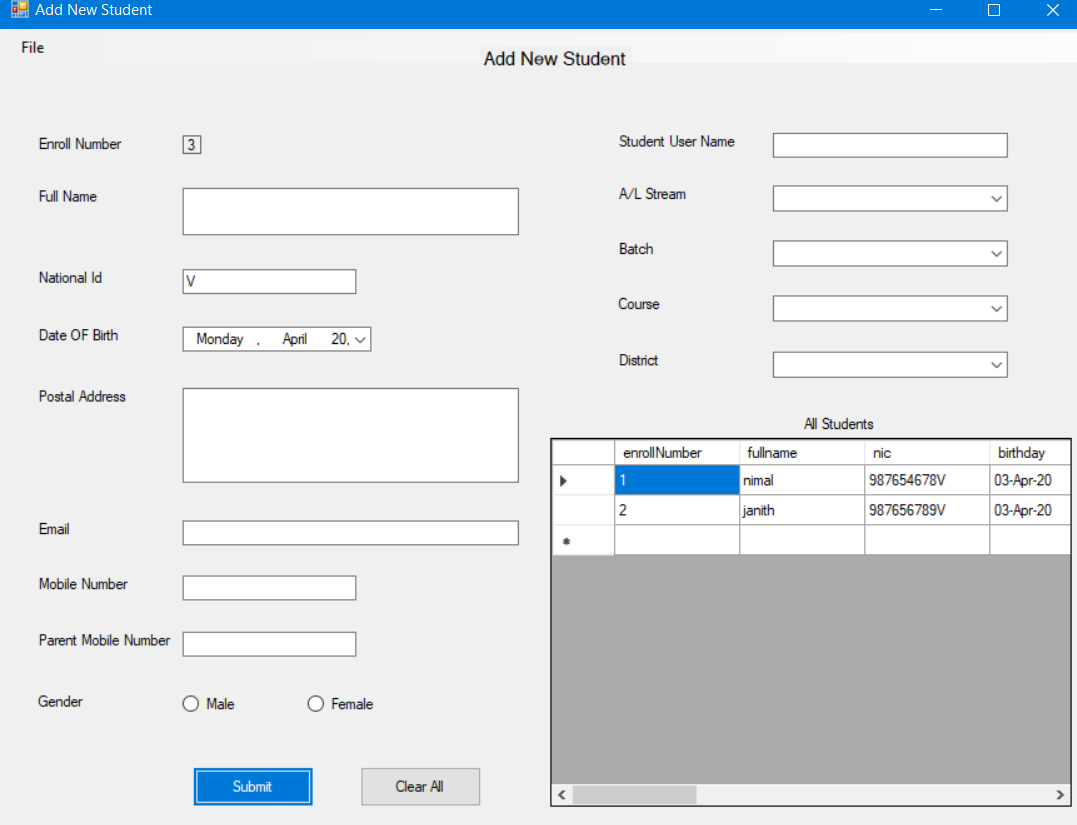
The simple concept of TDD is to write and correct the failed tests before writing new code (before development). This helps to avoid duplication of code as we write a small amount of code at a time in order to pass tests. (Tests are nothing but requirement conditions that we need to test to fulfill them). Test-Driven development is a process of developing and running automated test before actual development of the application. Hence, TDD is sometimes also called as Test First Development.

## 7.2Test cases

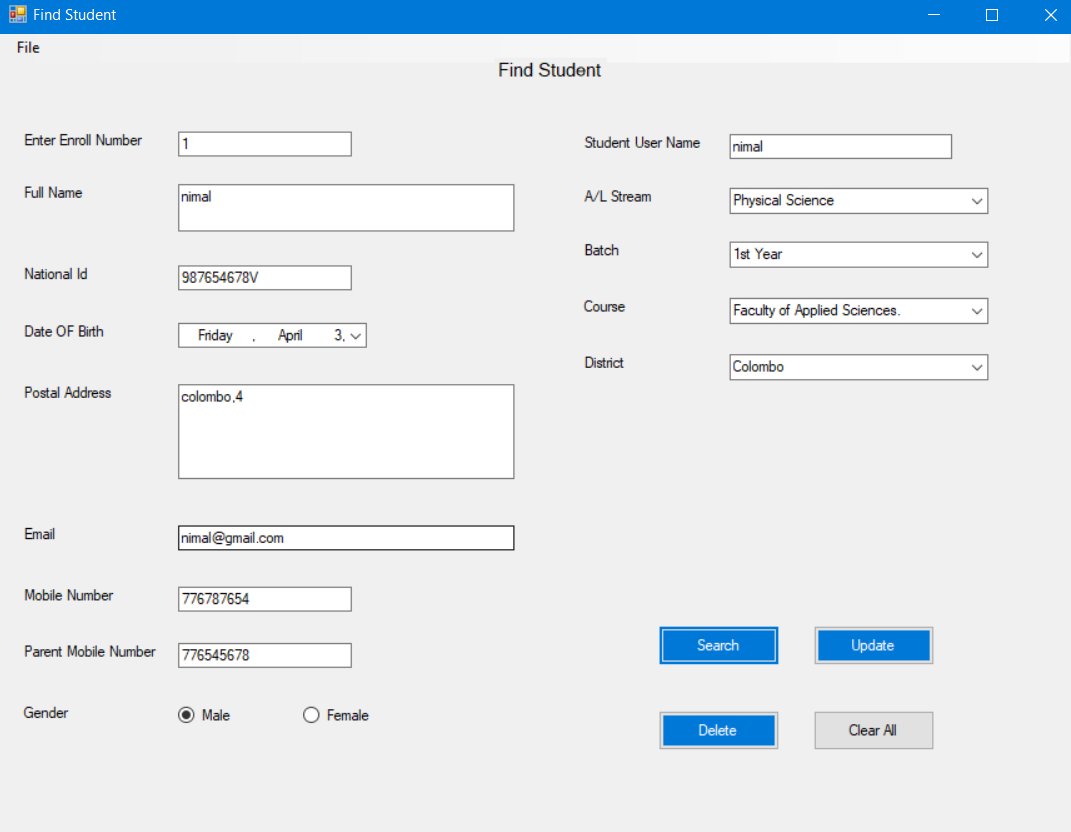


From this form we can add new student

As example lets enter new student details of Janith.

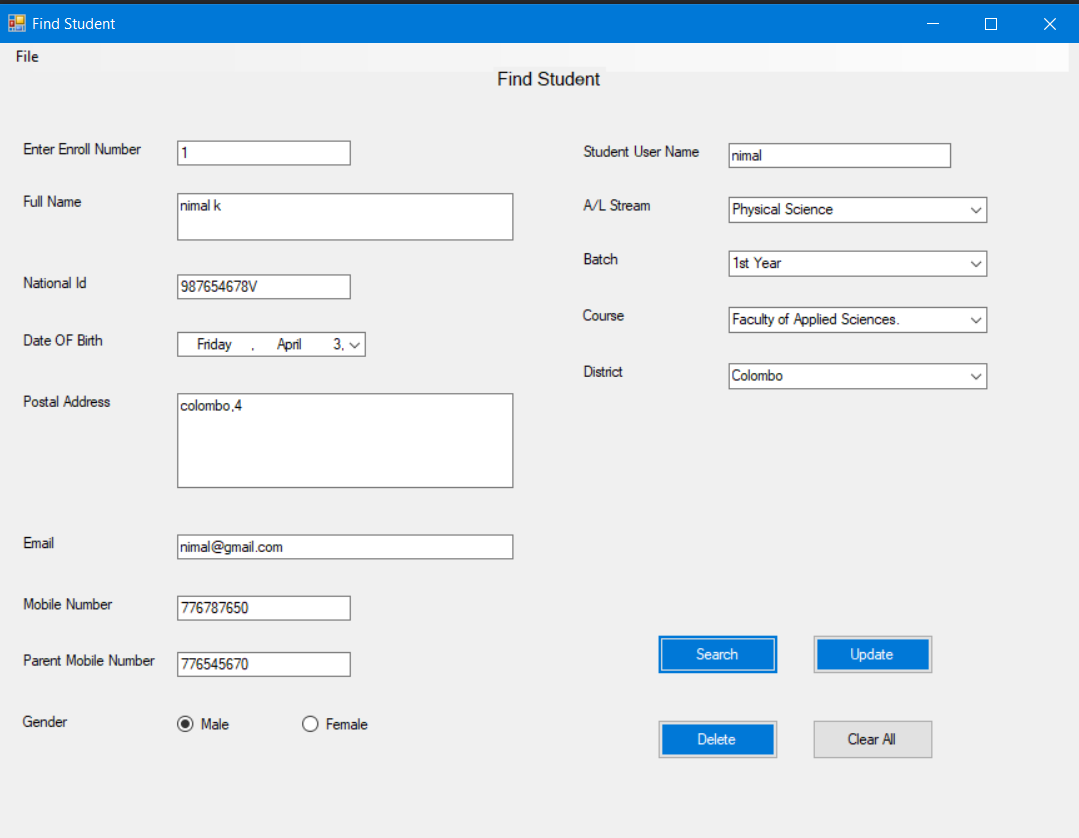


Now we can see the details of Janith were added into the table.

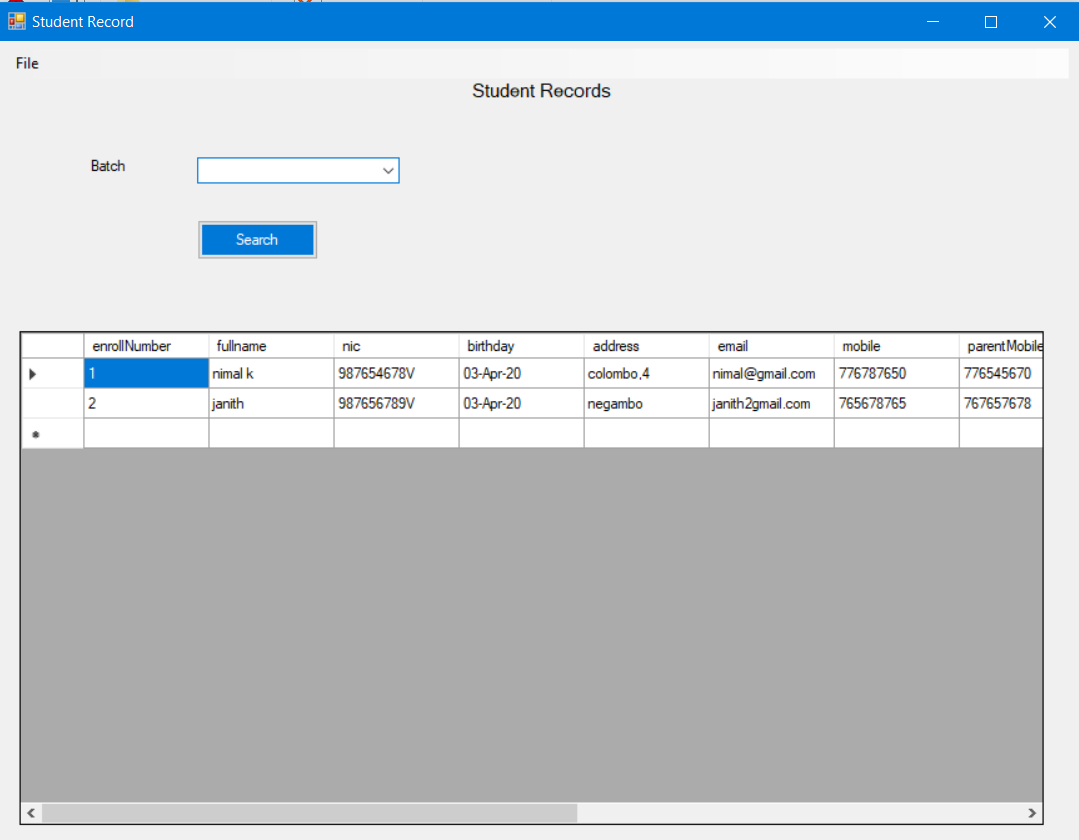


From this form can find student details and update student details

As example lets change name like “ nimal k”. At the beginning wants to enter student’s enroll number in the input field.

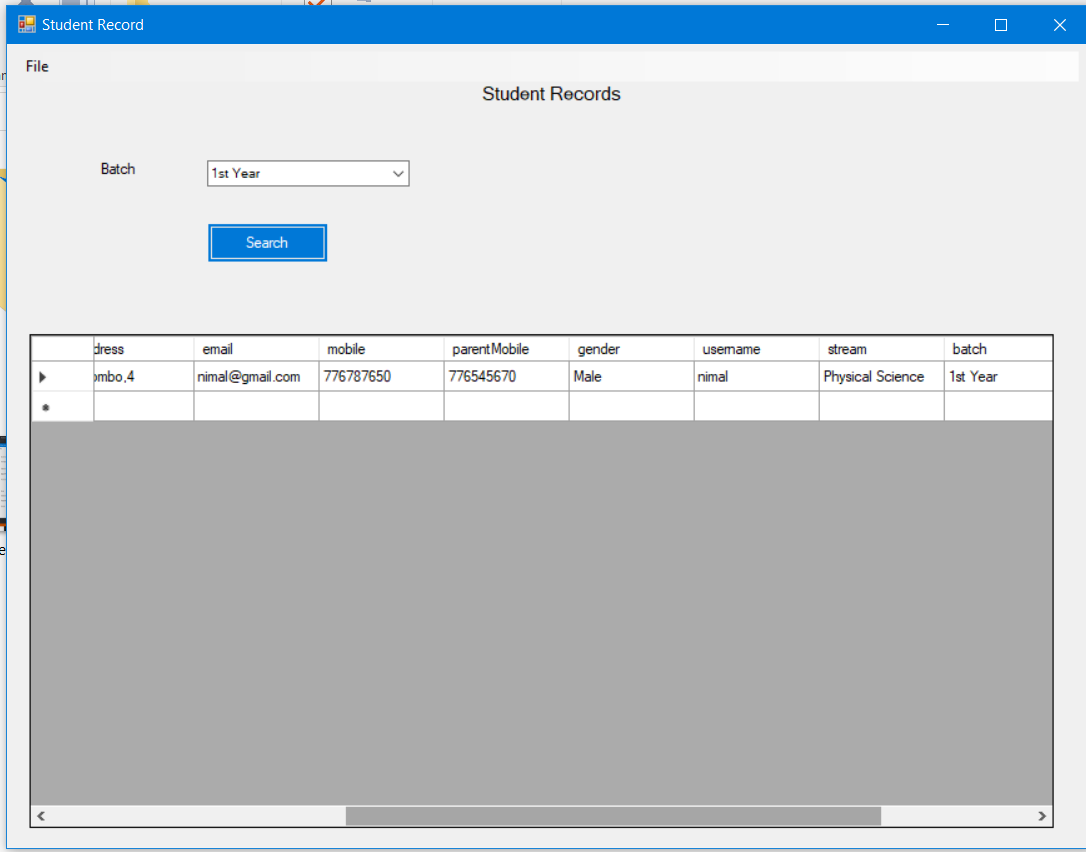


We can see the name of the student’s name was changed.

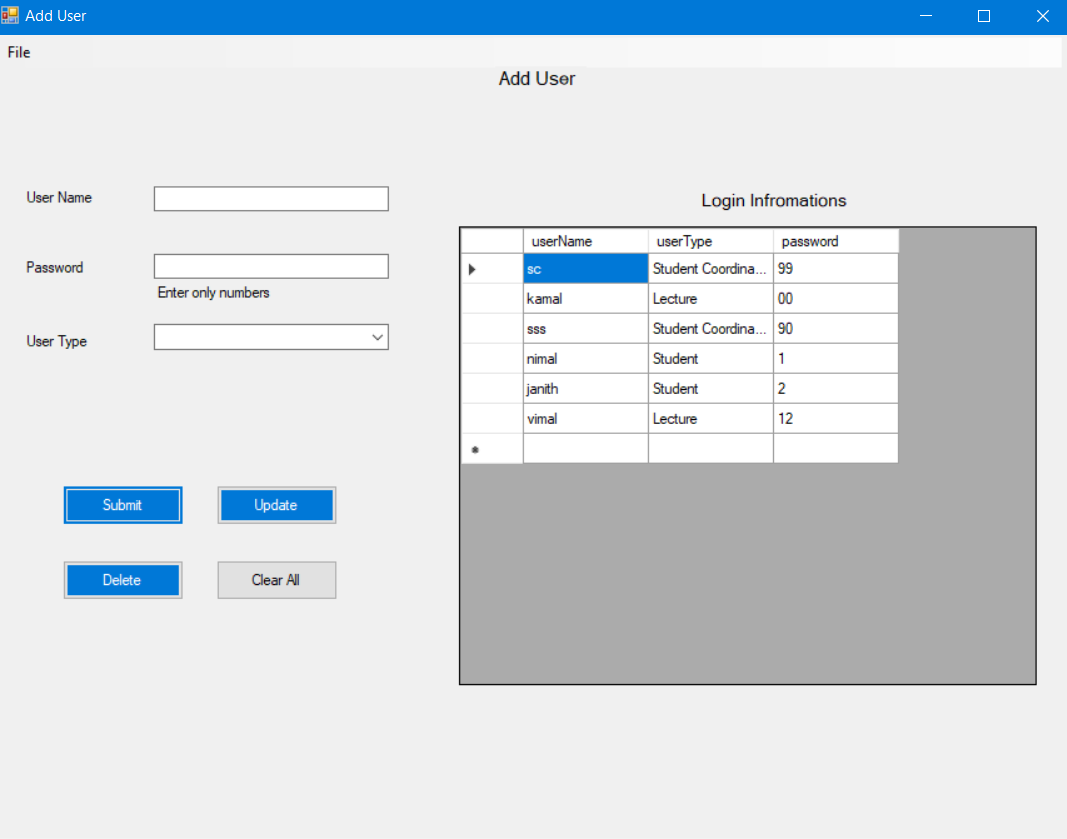


From this form can see all the student details and details according to the batch.

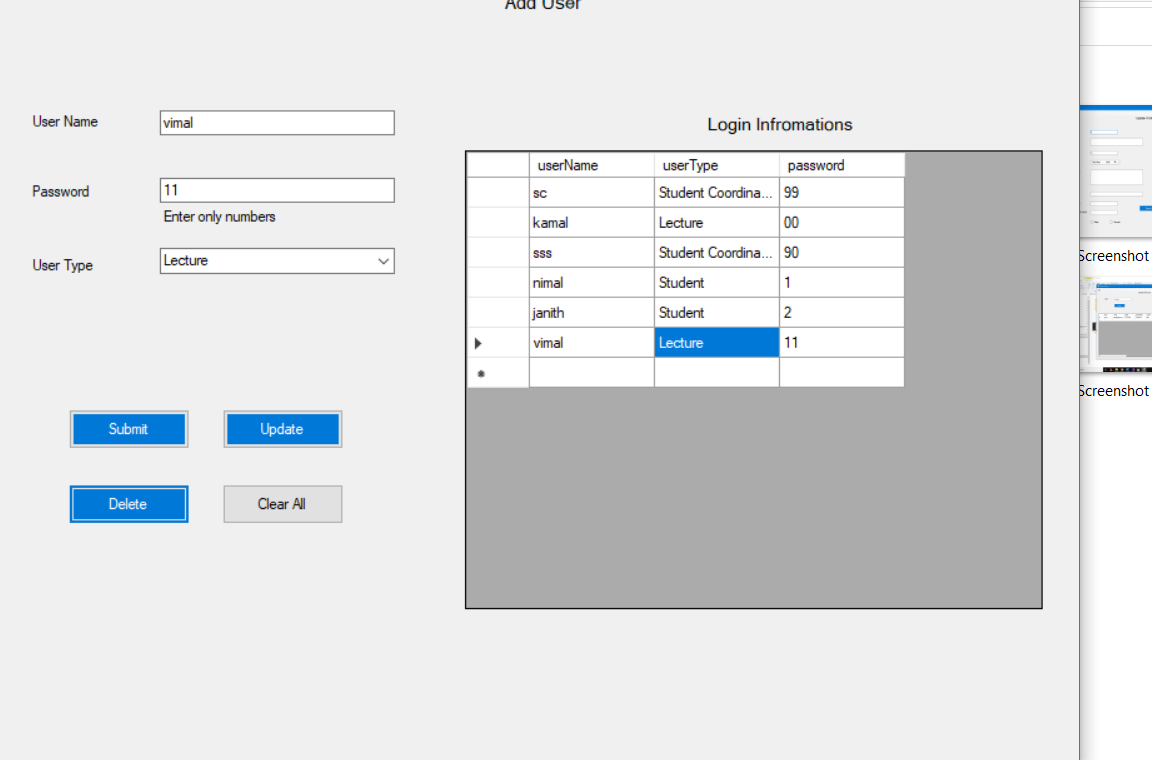
Let select batch at the combo box.



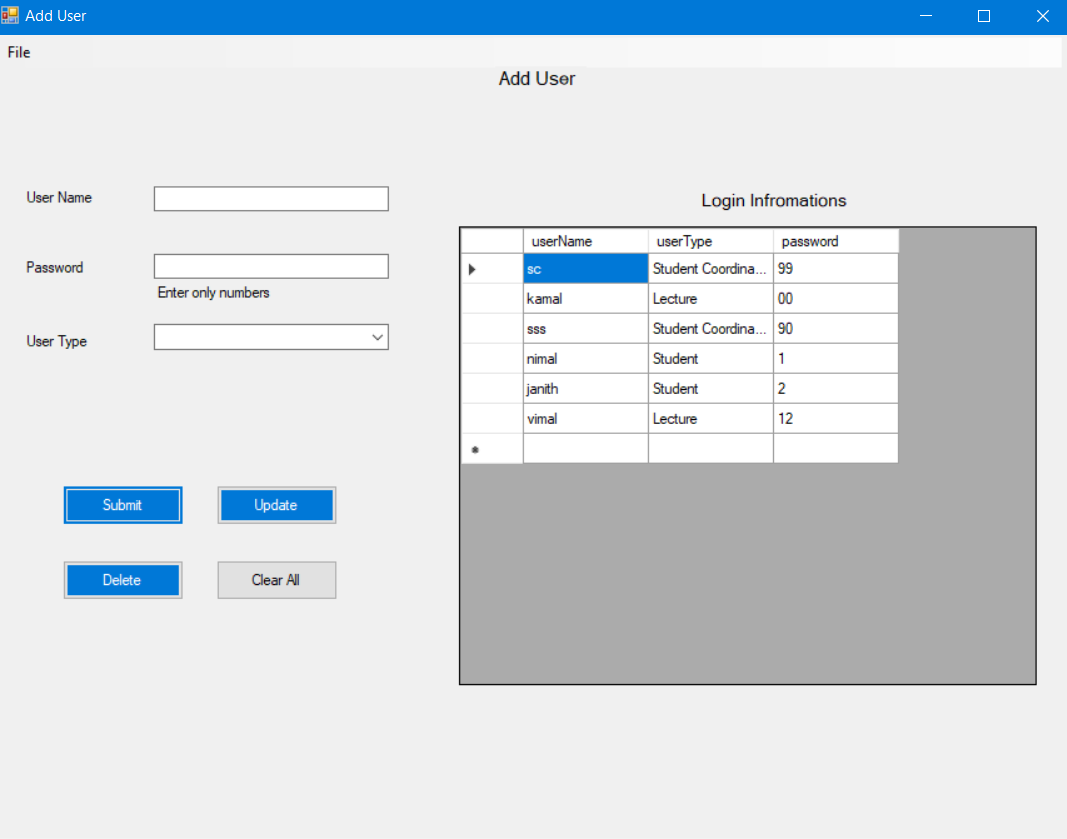
We can see only 1st year student details shown in the table.



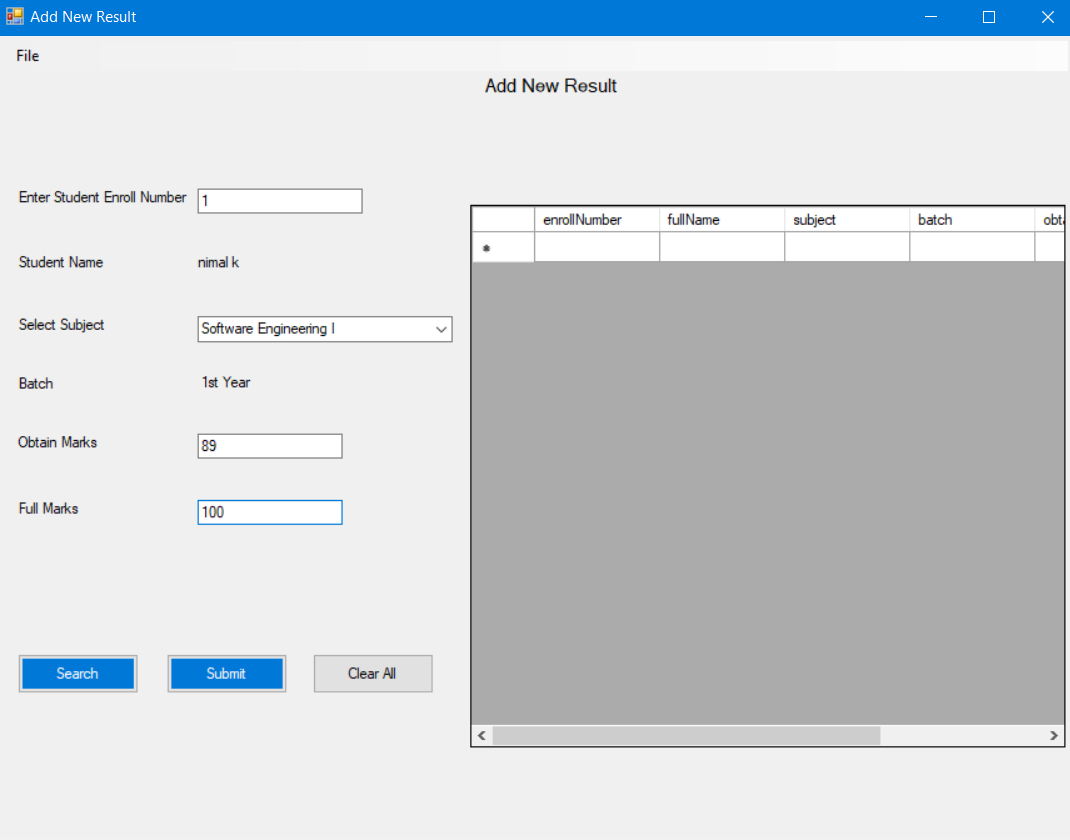
We can add new user into database from this from



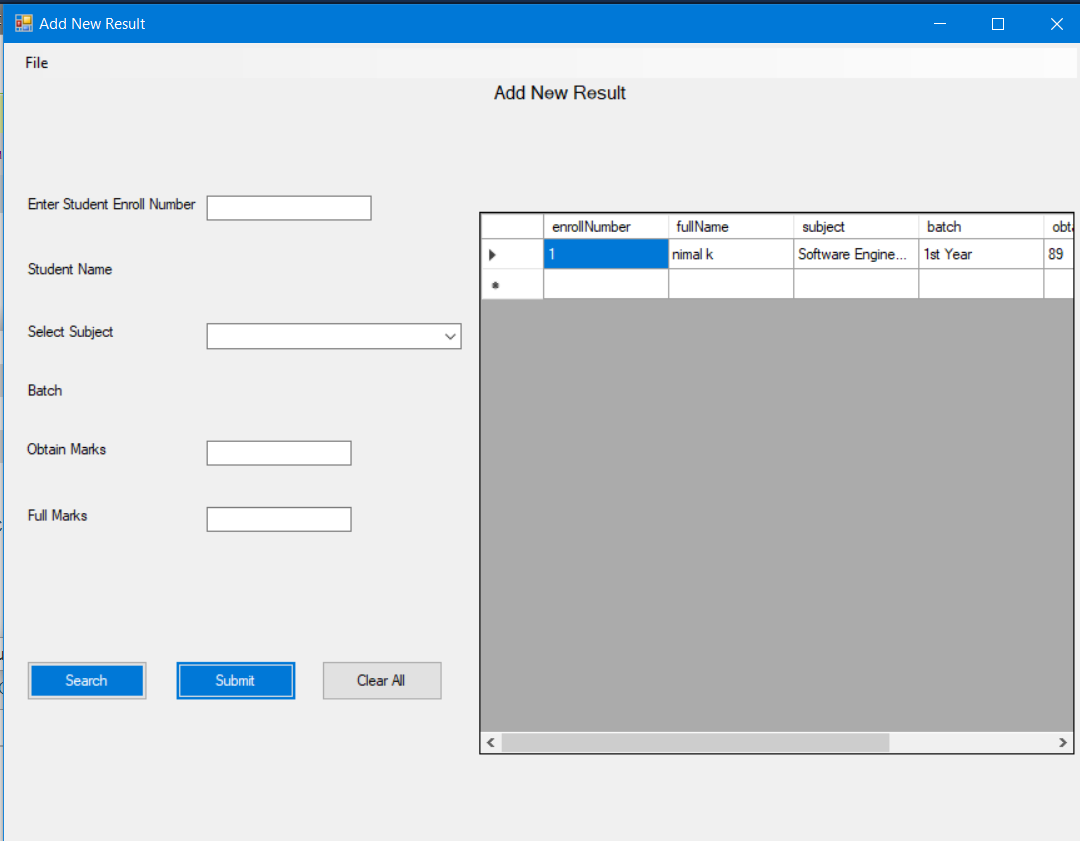
We can add user like Vimal into data base then can press submit button after that we can see the details of the Vimal can see in the table.



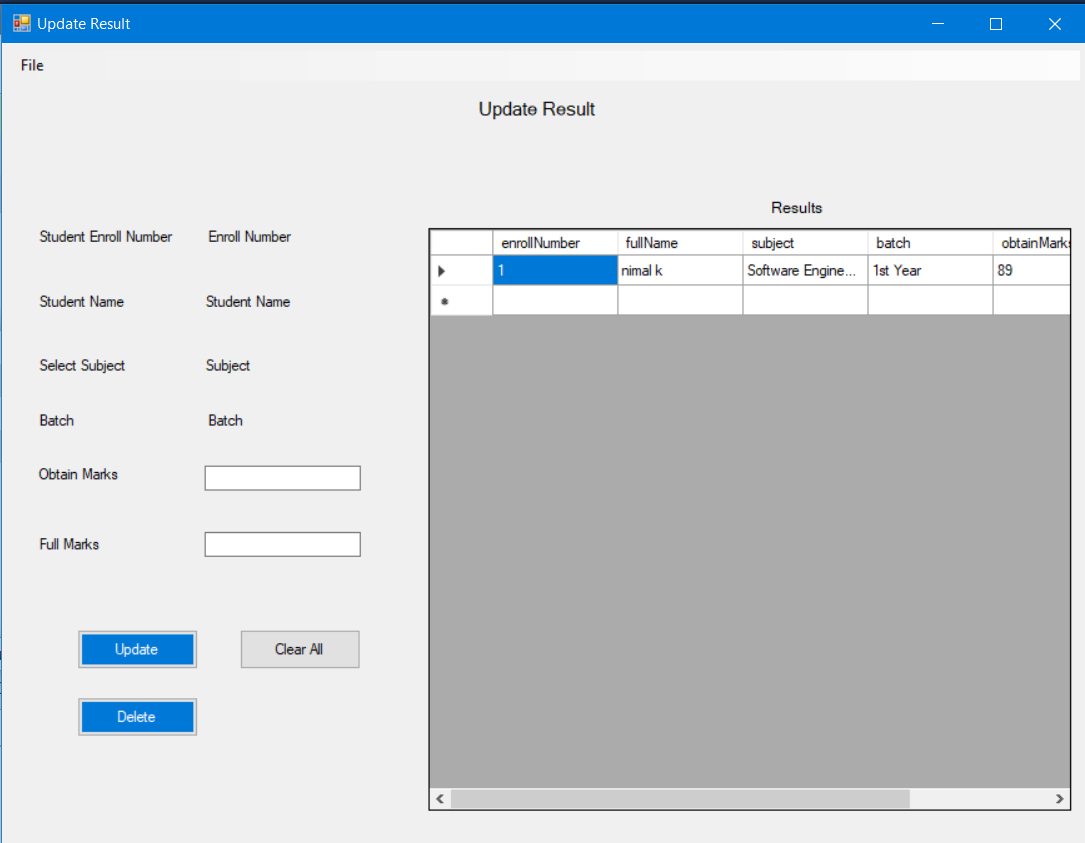
We can update the details of the user by pressing the table row and it appears in the textboxes then can update the details. After that we can the password updated like that.



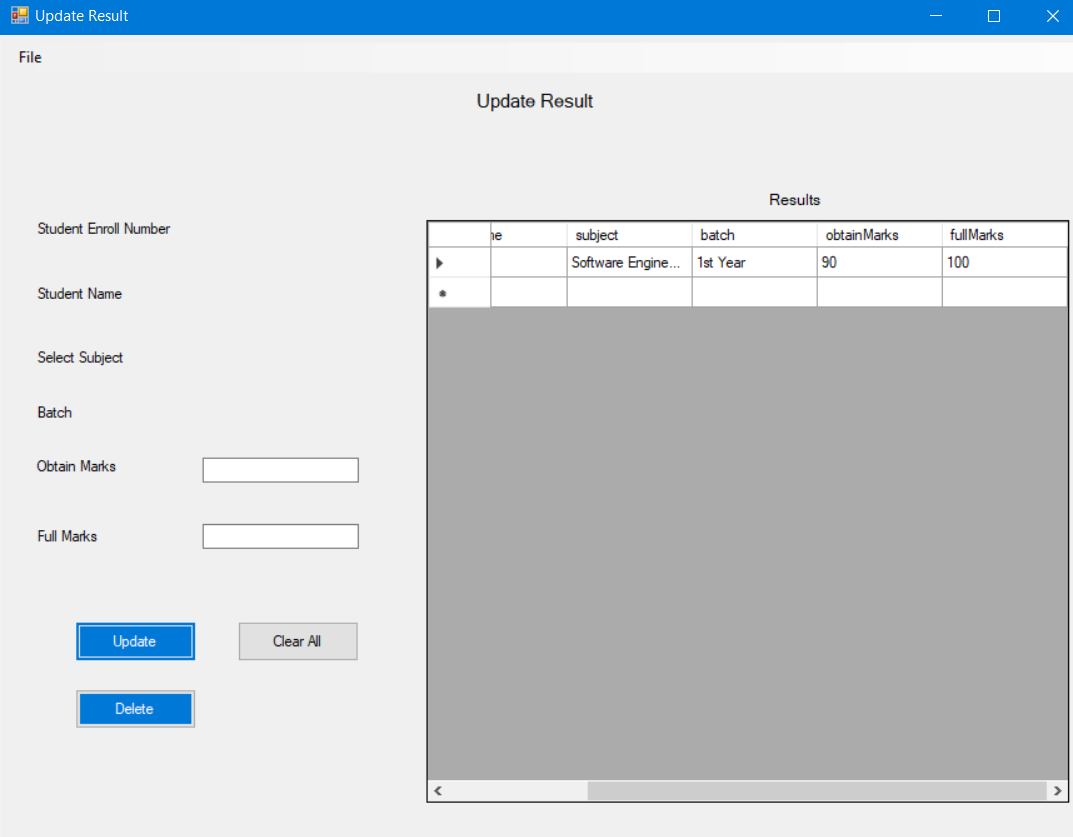
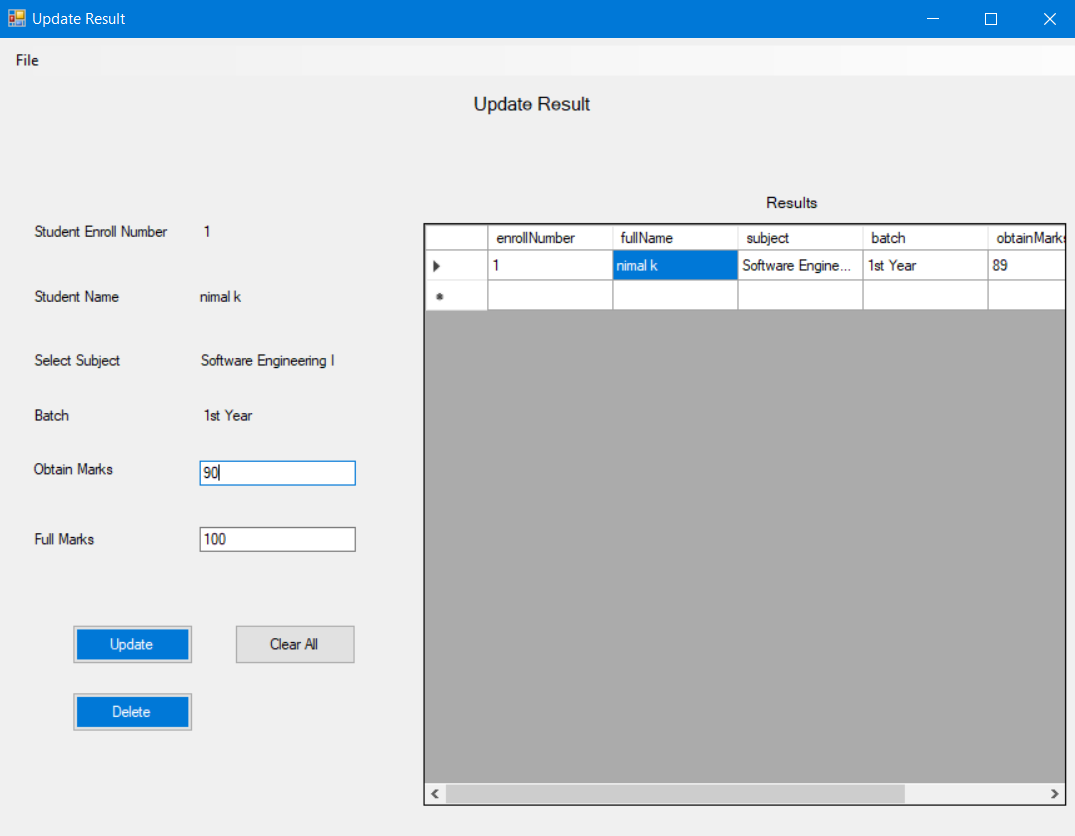
We can add new result of the student from this form by search the student enroll number.



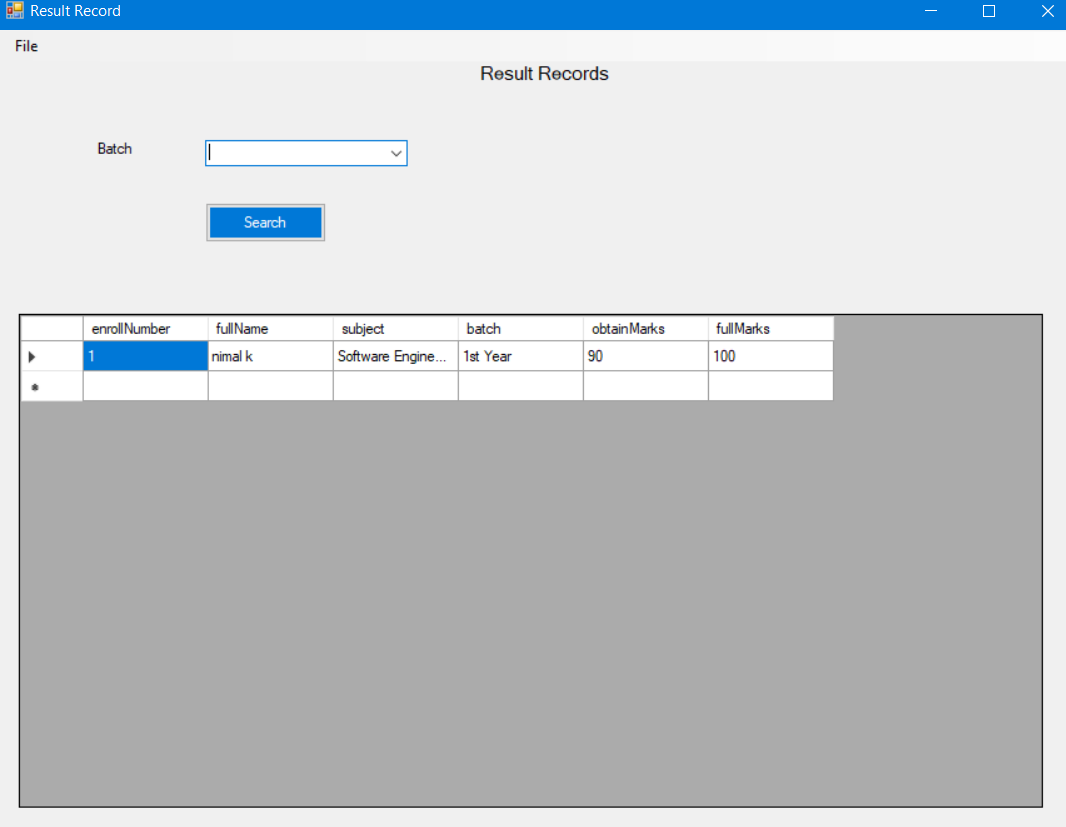
After that we can see the results details of the student shown in the table.



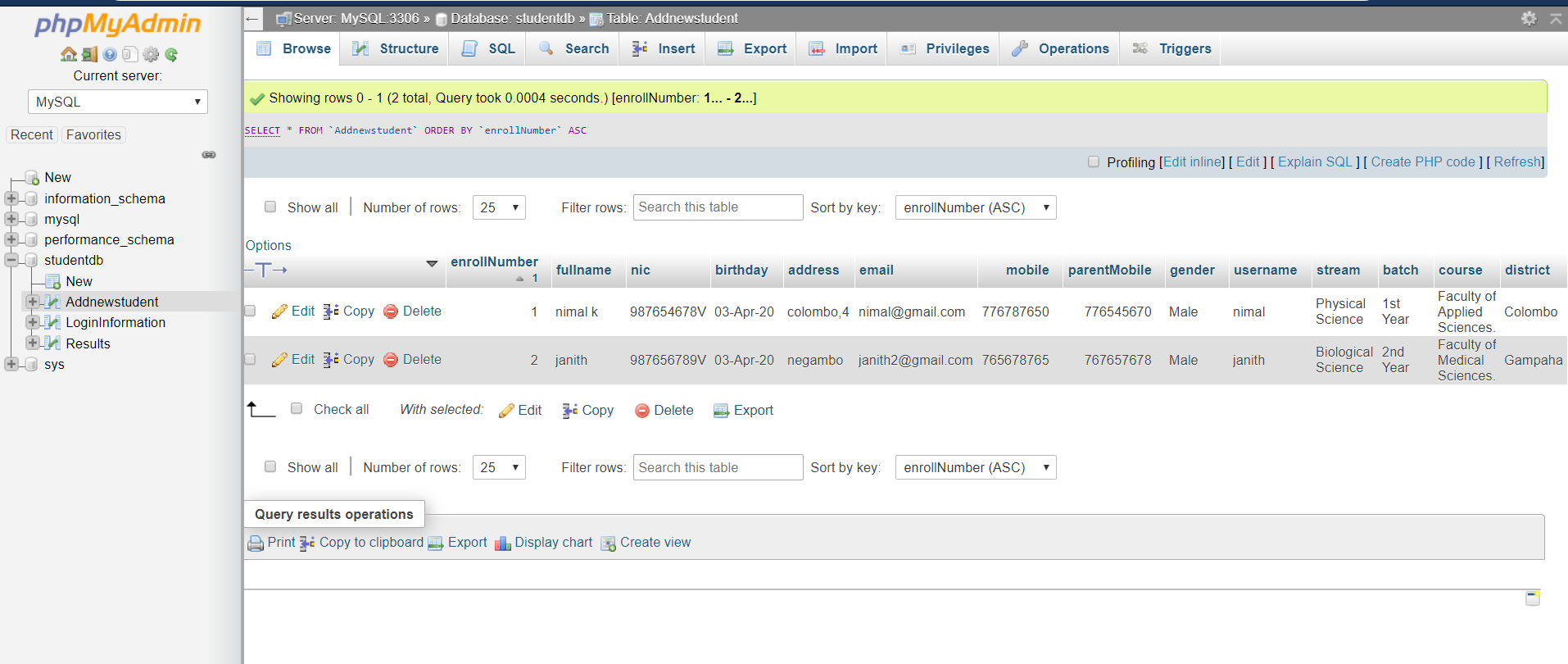
If we want to update the results can select the row of the result and the details of the selected row appear in the textboxes.

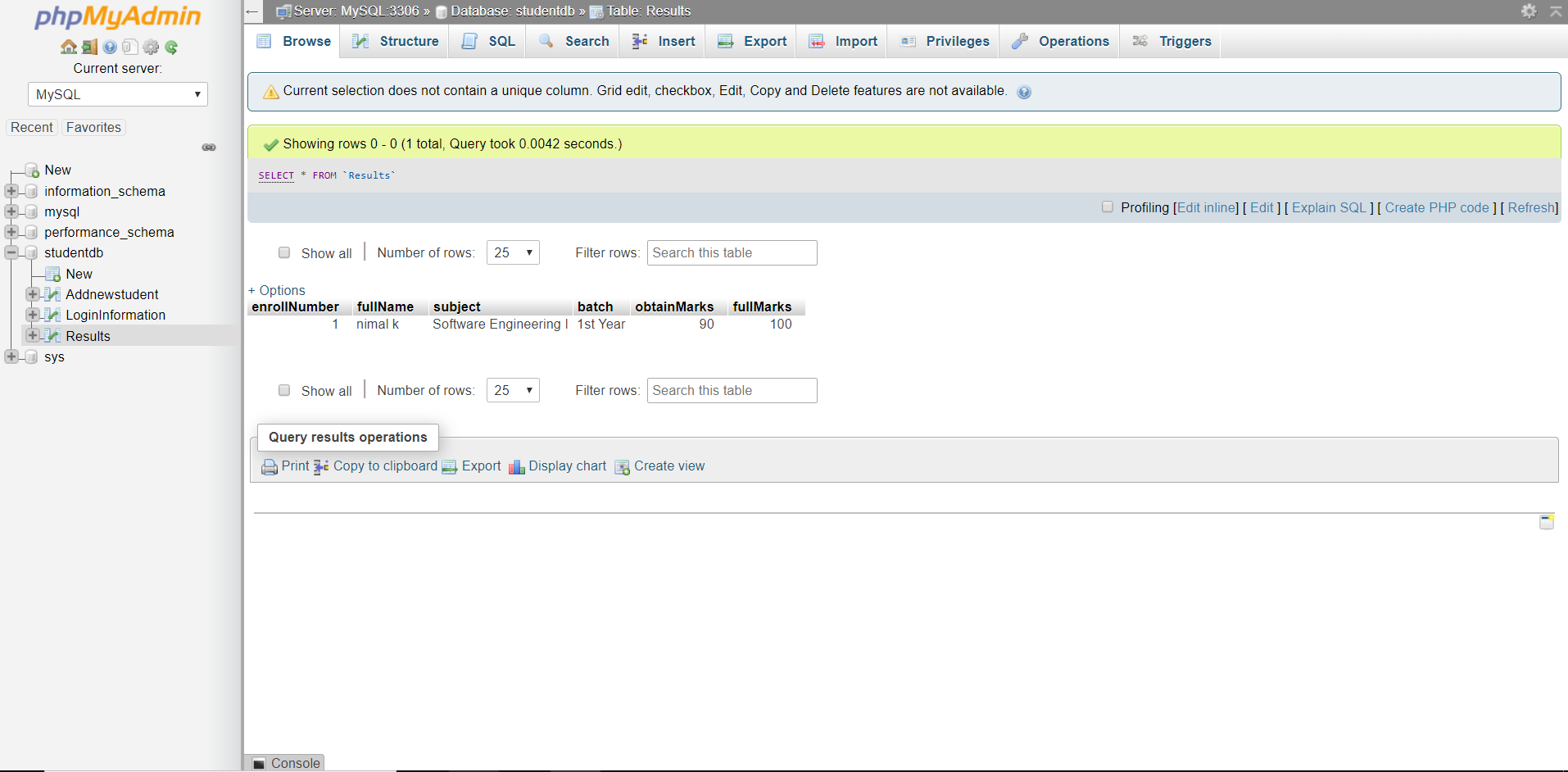


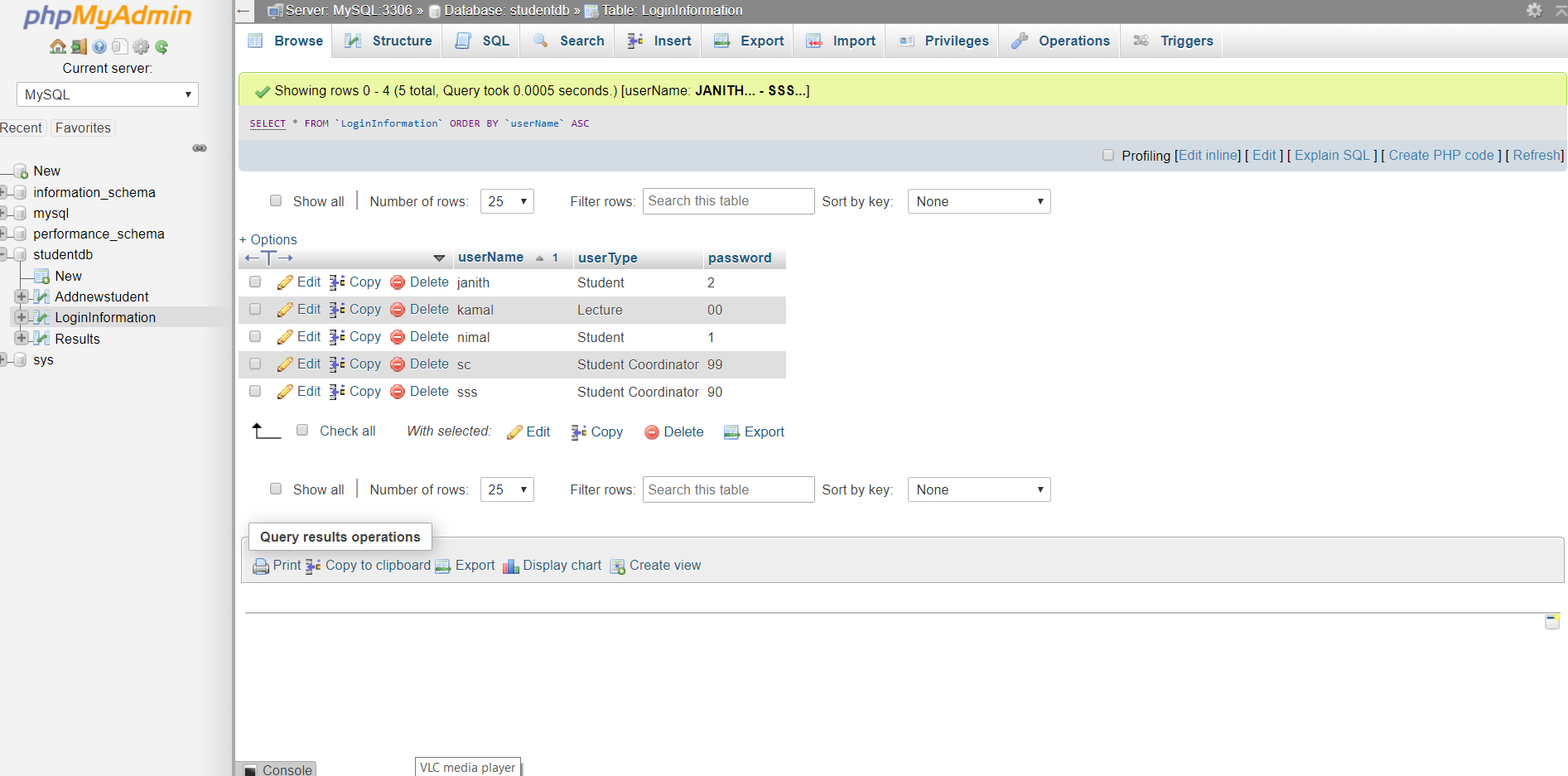
We can see the results was updated.



We can see all the results of the student in the table and can sort details as batch.

 Student details

Results details

Login details

# 8. Explanation of deployment techniques that are selected and critically evaluation of the suitability of the selection.

There are two categories of ASP.NET deployment:

**Local deployment:** In this case, the entire application is contained within a virtual directory and all the contents and assemblies are contained within it and available to the application.

**Global deployment:** In this case, assemblies are available to every application running on the server.

There are different techniques used for deployment, however, the following most common and easiest ways of deployment:

* XCOPY deployment
* Copying a Website
* Creating a set up project

## 8.1 XCOPY Deployment

XCOPY deployment means making recursive copies of all the files to the target folder on the target machine. Can use any of the commonly used techniques:

* FTP transfer
* Using Server management tools that provide replication on a remote site
* MSI installer application

XCOPY deployment simply copies the application file to the production server and sets a virtual directory there. Need to set a virtual directory using the Internet Information Manager Microsoft Management Console (MMC snap-in).

## 8.2 Copying a Website

The Copy Web Site option is available in Visual Studio. It is available from the Website -> Copy Web Site menu option. This menu item allows copying the current web site to another local or remote location. It is a sort of integrated FTP tool.

Using this option, connect to the target destination, select the desired copy mode:

* Overwrite
* Source to Target Files
* Sync UP Source and Target Projects

Then proceed with copying the files physically. Unlike the XCOPY deployment, this process of deployment is done from Visual Studio environment. However, there are following problems with both the above deployment methods:

* You pass on your source code.
* There is no pre-compilation and related error checking for the files.
* The initial page load will be slow.

## 8.3 Creating a Setup Project

In this case, deploy a Windows Forms Form application using the Setup Wizard method. Then I will create a simple installer for the application.