****

**Final Project**

|  |  |
| --- | --- |
| **Final Project – Assessment Sheet** | |
| **Programme** | Assured Diploma in Information Technology |
| **Assessor/Lecturer** | Mr. Manusha Jayarathne |
| **Unit** | Final Project |
| **Project Title** | DiTEC Final Project |
| **Student’s Name** | Lahiru Sadaruwan Kumara Karunarathna |
| **Student’s Registration No** | E161509 |
| **Submission Date** | 2023/02/17 |
| **Marks Achieved (%)** |  |
| **Assessor/Lecturer Feedback** |  |
|  |  |
|  |  |

**Important Points.**

* Carefully check the hand in date and the instructions given with the project. **Late submissions will not be accepted.**
* Ensure that sufficient time is spent to complete the assignment by the due date.
* Do not wait till the last minute to get feedback on the project. Such excuses will not be accepted for late submissions.
* You must be responsible for efficient management of your time.
* If you are unable to hand in your project on time and have valid reasons such as illness, you may apply (in writing) for an extension.
* Failure to achieve at least 40.00% marks will result in a Repeat grade.
* Non-submission of work without valid reasons will lead to an automatic Repeat. You will then be asked to complete an alternative project.
* If you use other people’s work or ideas in your assignment, it must be properly referenced, using the HARVARD referencing system, in your text or any bibliography. Otherwise, you’ll be found guilty of committing plagiarism.
* If you are caught plagiarising, your grade will be reduced to a Repeat or at worst, you could be excluded from the course.

**Project Guidelines**

* You need to **create a software** solution for Skills International School using **visual studio** as your final project.
* Use **C# Programming language** for your project
* Follow the structure given in this document. Design given forms and apply value additions for it according to your requirement.
* Software **should** contain **minimum** **two forms**. Additional forms and features will get additional marks. (Login form and one Registration form is compulsory)
* Please follow the given instructions before you start the project.

**Instructions**

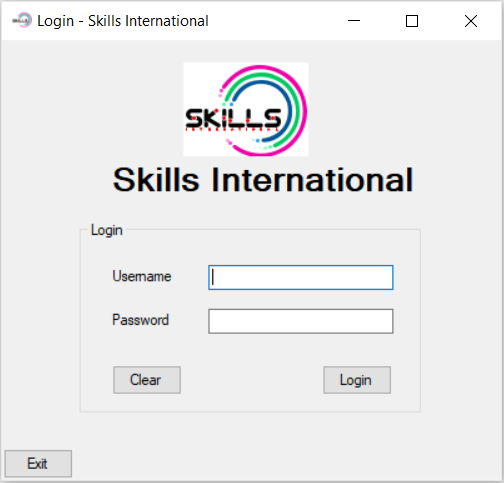
* Design the software system according to the given instructions.
* Use appropriate fonts, colors, and images for Frontend (GUI design) of your software.
* Proposed Software system **must have a database and properly connected** to the system. (Database should be designed using SQL Server Management Studio)
* Database must contain minimum **one Table** (Table column structure provided in Figure X).
* **Test the software** and submit the project on or before the given deadline.
* Create a **Microsoft Word Document** including all the screenshots of forms in your system and code snippets of each form.
* The entire project brief (These nine pages) should be attached as the first section of your project documentation.

**Word Processing Rules**

* The font size should be 12 point and should be in the style of Time New Roman.
* Set line spacing to 1.15. Justify all paragraphs.
* Ensure that all headings are consistent in terms of size and font style.
* Use the spell check and grammar check function of the word processing application to review the use of language on your assignment.

**Proposed Project**

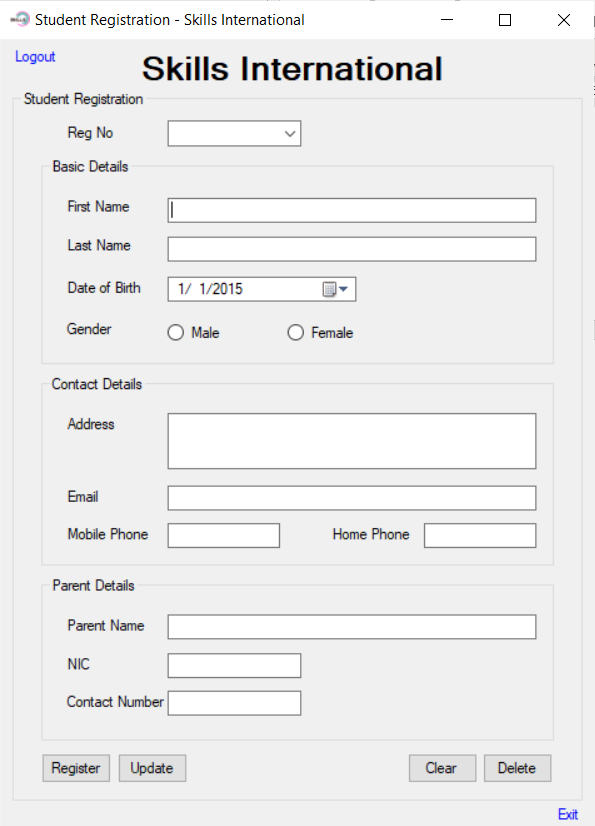
Figure I – Login Form



* Above form consist of;
  + One Picture Box
  + One Group Box
  + Three Labels
    - Skills International
    - Username
    - Password
  + Two Text Boxes
    - Username
    - Password
  + Three Buttons
    - Clear
    - Login
    - Exit

Design the above form using these tools and change respective properties using suitable values.

Figure II – Registration Form



* Above form consist of;
  + Four Group Boxes
    - Student Registration
    - Basic Details
    - Contact Details
    - Parent Details
  + Thirteen Labels
    - Skills International
    - Reg No
    - First Name
    - Last Name
    - Date of Birth
    - Gender
    - Address
    - Email
    - Mobile Phone
    - Home Phone
    - Parent Name
    - NIC
    - Contact No
  + Nine Text Boxes
    - First Name
    - Last Name
    - Address (Multiline)
    - Email
    - Mobile Phone
    - Home Phone
    - Parent Name
    - NIC
    - Contact No
  + Four Buttons
    - Register
    - Update
    - Clear
    - Delete
  + Two LinkLabels
    - Logout
    - Exit
  + Two Radio Buttons
    - Male
    - Female
  + One DateTimePicker
    - Date of Birth
  + One ComboBox
    - Reg No

Design the above form using these tools and change respective properties using suitable values.

**Project Instructions**

**Designing**

* Design the project using Microsoft Visual Studio. **Designing above two forms is a must**, you can add more forms based on your imagination and those will be considered when marking.
* **Change the properties** of each object with suitable values.

**Functional Operations**

* When user runs the project, Login form needs to be appeared.

**Form I – Login Form (Figure I)**

* There is **one user** in the system. User name is **“Admin”** & Password is **“Skills@123”**. When user click on Login ( ) button after applying “Admin” as the username and “Skills@123” as the Password, the system needs to **hide the login form** and **open the second form** which is Registration form (Figure II). If the login credentials are wrong (Either Username, either Password or both), system needs to generate an error message (Figure III).

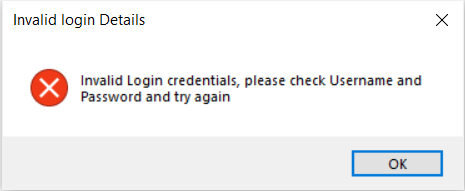


Figure III – Error Message

* You can **increase number of users** for the system. If so, there needs to be a table in the database to track those usernames and passwords as shown in figure IX.
* When user clicks on Clear ( ) button, texts in both username textbox and password textbox needs to be cleared and the username textbox needs to be **focused**.
* When you click on the Exit Button ( ) it needs to appear the following message box.

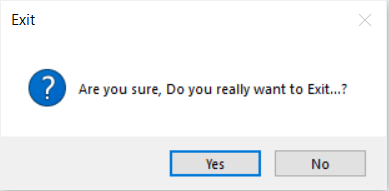


Figure IV – Exit Message

* If you click on Yes button, the entire system needs to be closed and if you click on No, the system stays without a change, but the dialogue box needs to be closed.

**Form II – Registration Form (Figure II)**

* The Form II (Figure II) is designated form for Student Registrations.
* Create the Database using Microsoft SQL Server Management Studio. Name the database as **Student**.
* Create the **Registration Table**.
* Connect the created Student Database with your C# project.
* In student Registration form, a registration can be done by clicking on Register Button ( ). Once the user clicks on **Register Button** after providing all the details in related fields in the registration form, the details need to be added in the “**Registration**” Table (Figure X) of “**Student**” Database and needs to appear following message box as well.

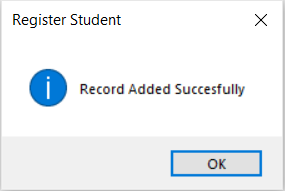


Figure V – Register Message Box



* Give the facility to the user to update records using **Update Butto**n ( ).
* Once the update executed, show the following message box.

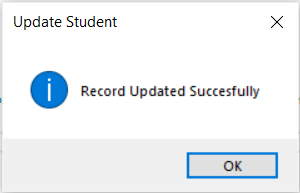


Figure VI – Update Message Box

* Give the facility for the user to Delete a selected record from the Database Table using the **Delete Button** ( ). Before Execution of Delete function, need to get users confirmation to delete. For that, use following message box or similar one.

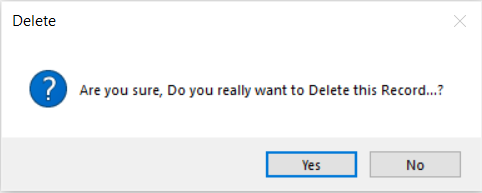


Figure VII – Delete Message Box

* Once the user clicks on Yes from the appearing message box, the details needs to be deleted from the system database based on the Registration Number of the record. After the execution of the Delete, following message box needs to be appeared.

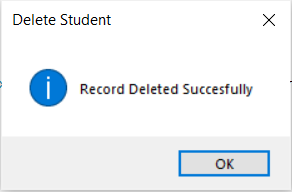


Figure VIII – Delete Confirmation Message Box



* When the user clicks on the **Reg No dropdown list** ( ), the user can see the registration numbers (Previously registered) appearing in the drop down list. When the user selects a registration number, it needs to work as a **Search** function when user changes the text of this and need to fill all other fields in the form by focusing the Reg No from the Database Table.
* When you click on the Exit Link Label ( ) it needs to appear the Exit message box (Figure IV).
* When user clicks on Logout Link Label ( ), it needs to logout from the system, disappear the Registration form and appear the login form (Figure I).

**Database Information**

Set Database name as “**Student**” and Registration Table (Figure X) needs to be there inside.

When designing the **Registration** Table in the Database, apply **regNo** column as the **primary key** for the table.



Figure IX – Database Table - **Logins**



Figure X – Database Table - **Registration**

**Table Structure**

Use the following table to identify data types for each column when you create the “**Registration”** table.

|  |  |
| --- | --- |
| **Field Name** | **Datatype** |
| regNo | Integer |
| firstName | varchar(50) |
| lastName | varchar(50) |
| dateOfBirth | dateTime |
| gender | varchar(50) |
| address | varchar(50) |
| email | varchar(50) |
| mobilePhone | Integer |
| homePhone | Integer |
| parentName | varchar(50) |
| nic | varchar(50) |
| contactNo | Integer |

# Acknowledgement

I would like to express my heartfelt thanks to the lecturers who contributed to the success of my final project "Skills International" software.

Thanks for all your encouragement!

# Scope Statement

The "Skills International" Software project aims to create a software solution for international schools that will streamline various administrative tasks, such as student registration and staff management. The software will include several key features, including a Login Form, Home Form, Student Register Form, Manage Staff Form, and Manage Users Form etc.

The project will begin with a detailed analysis of the requirements for the software, including input from school administrators and staff. The software will be designed to meet the specific needs of international schools, with a focus on ease of use and flexibility.

The Login Form will allow users to access the software securely and will be designed to prevent unauthorized access. The Home Form will provide a dashboard view of important information, such as upcoming events and announcements. The Student Register Form will allow for easy and efficient student registration, while the Manage Staff Form will allow school administrators to manage their staff members' information and schedules. The Manage Users Form will enable administrators to manage user accounts and permissions.

Table of Contents

[Acknowledgement 10](#_Toc127563222)

[Scope Statement 11](#_Toc127563223)

[List of Tables 14](#_Toc127563224)

[List of Figures 14](#_Toc127563225)

[1. Database Structure 16](#_Toc127563226)

[1.1 Description 16](#_Toc127563227)

[1.2. StudentRegistration Table Structure 16](#_Toc127563228)

[1.3. Staff Table Structure 16](#_Toc127563229)

[1.4. Users Table Structure 17](#_Toc127563230)

[1.5. Screenshot of Tables Structures 17](#_Toc127563231)

[2. Loading Form 18](#_Toc127563232)

[2.1. Description (How It’s Work) 18](#_Toc127563233)

[2.2. Screenshot of Loading Screen 18](#_Toc127563234)

[3. SQL Connection Class 19](#_Toc127563235)

[3.1. Description 19](#_Toc127563236)

[4. Login Form 20](#_Toc127563237)

[4.1. Description 20](#_Toc127563238)

[4.2. Screenshot of Login Form 20](#_Toc127563239)

[4.3. Screenshot of Wrong Credential Message Box 21](#_Toc127563240)

[4.4. Screenshot of Exit Button Handle Click 21](#_Toc127563241)

[5. Home Form 23](#_Toc127563242)

[5.1. Description 23](#_Toc127563243)

[5.2. Screenshot of Home Form 23](#_Toc127563244)

[6. Student Registration Form 25](#_Toc127563245)

[6.1. Description 25](#_Toc127563246)

[6.2. Screenshot of Student Registration Form 25](#_Toc127563247)

[6.3. Registration Button Click Event Code Explanation 26](#_Toc127563248)

[6.4. DataGridView Click Event Description 28](#_Toc127563249)

[6.5. Explanation of Update Button Click Event 29](#_Toc127563250)

[6.6. Explanation of Delete Button Click Event 31](#_Toc127563251)

[6.7. Explanation of Clear Button Click Event 32](#_Toc127563252)

[7. Manage Staff Form 33](#_Toc127563253)

[7.1. Description 33](#_Toc127563254)

[7.2. Screenshot of Manage Staff Form 33](#_Toc127563255)

[7.3. Explanation of Add Button Click Event 34](#_Toc127563256)

[7.4. Add Button Click Event Code Snippet 35](#_Toc127563257)

[7.5. DataGridView Click Event Description 36](#_Toc127563258)

[7.6. Explanation of Edit Button Click Event 37](#_Toc127563259)

[7.7. Explanation of Delete Button Click Event 38](#_Toc127563260)

[7.8. Explanation of Clear Button Click Event 40](#_Toc127563261)

[8. Users Form 41](#_Toc127563262)

[8.1. Description 41](#_Toc127563263)

[8.2. Screenshot of Users Form 41](#_Toc127563264)

[8.3. 41](#_Toc127563265)

[8.3. Explanation of Add Button Click Event 42](#_Toc127563266)

[8.4. DataGridView Click Event Description 44](#_Toc127563267)

[8.5. Explanation of Edit Button Click Event 45](#_Toc127563268)

[8.6. Explanation of Delete Button Click Event 46](#_Toc127563269)

[8.7. Explanation of Clear Button Click Event 48](#_Toc127563270)

# List of Tables

[Table 1 Student Registration Table Structure 16](#_Toc127561644)

[Table 2 Staff Table Structure 16](#_Toc127561645)

[Table 3 Users Table Structure 17](#_Toc127561646)

# List of Figures

[Figure 1 Database Sructure 17](#_Toc127560638)

[Figure 2 Loading Screen 18](#_Toc127560639)

[Figure 3 Loading Form Code Snippet 19](#_Toc127560640)

[Figure 4 Connection Class Code Snippet 19](#_Toc127560641)

[Figure 5 Login Screen 20](#_Toc127560642)

[Figure 6 Login Wrong Credentials Message 21](#_Toc127560643)

[Figure 7 Login - Exit Confirmation Message 21](#_Toc127560644)

[Figure 8 Login - Import Connection Class Snippet 21](#_Toc127560645)

[Figure 9 .Login - Exit Button Click Event Code Snippet 22](#_Toc127560646)

[Figure 10 Login Button Click Event Code Snippet 22](#_Toc127560647)

[Figure 11 Login - Password Show/Hide Code Snippet 22](#_Toc127560648)

[Figure 12 Login - Key Down Events Code Snippet 22](#_Toc127560649)

[Figure 13 Home Screen 23](#_Toc127560650)

[Figure 14 Home Form Closed Event Code Snippet 23](#_Toc127560651)

[Figure 15 Home - Navigation Button Click Events Code Snippet 24](#_Toc127560652)

[Figure 16 Home - Logout Confirmation Message 24](#_Toc127560653)

[Figure 17 Home - Logout Event Code Snippet 24](#_Toc127560654)

[Figure 18 Student Registration Screen 25](#_Toc127560655)

[Figure 19 Student Registration Form Closed Event Code Snipped 25](#_Toc127560656)

[Figure 20 Import Connection Class Code Snippet & creates a new instance of the “SqlCommand” class Code Snippet 25](#_Toc127560657)

[Figure 21 Student Registration Form Load Event Code Snippet 26](#_Toc127560658)

[Figure 22 Search and all Data load Custom Method Code Snipped 26](#_Toc127560659)

[Figure 23 Clear Custom Method Snipped 26](#_Toc127560660)

[Figure 24 Student Registration - Screenshot of Register Details Missing Warning Message Box 27](#_Toc127560661)

[Figure 25 Student Registration - Screenshot of Register Successful Message Box 27](#_Toc127560662)

[Figure 26 Student Registration - Register Button Click Event Code Snippet Part -1 27](#_Toc127560663)

[Figure 27 Student Registration - Register Button Click Event Code Snippet Part -2 28](#_Toc127560664)

[Figure 28 Student Registration - Screenshot of DataGridView Click Event 28](#_Toc127560665)

[Figure 29 Student Registration - DataGridView Click Event Code Snippet 29](#_Toc127560666)

[Figure 30 Student Registration - Screenshot of Update Warning Message Box 29](#_Toc127560667)

[Figure 31 Student Registration - Screenshot of Updated Successful Message Box 30](#_Toc127560668)

[Figure 32 Student Registration - Update Button Click Event Code Snippet Part -1 30](#_Toc127560669)

[Figure 33 Student Registration - Update Button Click Event Code Snippet Part -2 30](#_Toc127560670)

[Figure 34 Student Registration - Screenshot of Delete Confirmation Dialog Box 31](#_Toc127560671)

[Figure 35 Student Registration - Screenshot of Record Delete Successful Message Box 31](#_Toc127560672)

[Figure 36 Student Registration - Delete Button Click Event Code Snippet 32](#_Toc127560673)

[Figure 37 Student Registration - Clear Button Code Snippet 32](#_Toc127560674)

[Figure 38 Screenshot of Search Student 32](#_Toc127560675)

[Figure 39 Student Registration - Search Text Change Event Code Snippet 32](#_Toc127560676)

[Figure 40 Screenshot of Manage Staff Form 33](#_Toc127560677)

[Figure 41 Manage Staff Form Closed Event Code Snipped 33](#_Toc127560678)

[Figure 42 Import Connection Class Code Snippet & creates a new instance of the “SqlCommand” class Code Snippet 33](#_Toc127560679)

[Figure 43 Manage Staff Form Load Event Code Snippet 34](#_Toc127560680)

[Figure 44 Manage Staff - Search and all Data load Custom Method Code Snipped 34](#_Toc127560681)

[Figure 45 Manage Staff - Clear Custom Method Snipped 34](#_Toc127560682)

[Figure 46 Manage Staff - Add Button Click Event Code Snippet Part 1 35](#_Toc127560683)

[Figure 47 Manage Staff - Add Button Click Event Code Snippet Part 2 35](#_Toc127560684)

[Figure 48 Manage Staff - Screenshot of Record Added Successful Message Box 36](#_Toc127560685)

[Figure 49 Manage Staff - Screenshot of Record Adding Warning Message Box 36](#_Toc127560686)

[Figure 50 Manage Staff - Staff DataGrid Click Event Code Snippet 36](#_Toc127560687)

[Figure 51 Manage Staff - Screenshot of Staff DataGrid Click Event 37](#_Toc127560688)

[Figure 52 Manage Staff - Screenshot of Editing Warning Message Box 37](#_Toc127560689)

[Figure 53 Manage Staff - Screenshot of Edited Successful Message Box 38](#_Toc127560690)

[Figure 54 Manage Staff - Edit Button Click Event Code Snippet Part 1 38](#_Toc127560691)

[Figure 55 Manage Staff - Edit Button Click Event Code Snippet Part 2 38](#_Toc127560692)

[Figure 56 Manage Staff - Screenshot of Delete Confirmation Dialog Box 39](#_Toc127560693)

[Figure 57 Manage Staff - Screenshot of Record Delete Successful Message Box 39](#_Toc127560694)

[Figure 58 Manage Staff - Delete Button Click Event Code Snippet 39](#_Toc127560695)

[Figure 59 Manage Staff - Clear Button Code Snippet 40](#_Toc127560696)

[Figure 60 Manage Staff - Screenshot of Search Student 40](#_Toc127560697)

[Figure 61 Manage Staff - Search Text Change Event Code Snippet 40](#_Toc127560698)

[Figure 62 Manage Users Form Screen 41](#_Toc127560699)

[Figure 63 Users Form Closed Event Code Snipped 41](#_Toc127560700)

[Figure 64 Import Connection Class Code Snippet & creates a new instance of the “SqlCommand” class Code Snippet 41](#_Toc127560701)

[Figure 65 Manage Users - Users Form Load Event Code Snippet 42](#_Toc127560702)

[Figure 66 Manage Users - Data load Custom Method Code Snipped 42](#_Toc127560703)

[Figure 67 Manage Users - Clear Custom Method Snipped 42](#_Toc127560704)

[Figure 68 Manage Users - Add Button Click Event Code Snippet Part 1 43](#_Toc127560705)

[Figure 69 Manage Users - Add Button Click Event Code Snippet Part 2 43](#_Toc127560706)

[Figure 70 Manage Users - Screenshot of Record Added Successful Message Box 43](#_Toc127560707)

[Figure 71 Manage Users - Screenshot of Record Adding Warning Message Box 44](#_Toc127560708)

[Figure 72 Manage Users - Users DataGrid Click Event Code Snippet 44](#_Toc127560709)

[Figure 73 Manage Users - Screenshot of Users DataGrid Click Event 44](#_Toc127560710)

[Figure 74 Manage Users - Screenshot of Editing Warning Message Box 45](#_Toc127560711)

[Figure 75 Manage Users - Screenshot of Edited Successful Message Box 45](#_Toc127560712)

[Figure 76 Manage Users - Edit Button Click Event Code Snippet Part 1 46](#_Toc127560713)

[Figure 77 Manage Users - Edit Button Click Event Code Snippet Part 2 46](#_Toc127560714)

[Figure 78 Manage Users - Screenshot of Delete Confirmation Dialog Box 47](#_Toc127560715)

[Figure 79 Manage Users - Screenshot of Record Delete Successful Message Box 47](#_Toc127560716)

[Figure 80 Manage Users - Delete Button Click Event Code Snippet 47](#_Toc127560717)

[Figure 81 Manage Users - Clear Button Code Snippet 48](#_Toc127560718)

# 1. Database Structure

## 1.1 Description

## 

Database Name: “Skills\_International”

Tables Names:

* StudentRegistration
* Staff
* Users

“Skills\_International” is a database that contains information related to student registrations, staff members, and system users. It is designed to help manage and maintain the information associated with these entities, providing an efficient and organized way to store and access data.

## 1.2. StudentRegistration Table Structure

Table Student Registration Table Structure

|  |  |
| --- | --- |
| **Field Name** | **Datatype** |
| RegNo | int |
| FirstName | varchar(50) |
| LastName | varchar(50) |
| DateOfBirth | date |
| Gender | varchar(10) |
| Address | varchar(50) |
| Email | varchar(20) |
| MobilePhone | int |
| HomePhone | int |
| ParentName | varchar(50) |
| NIC | varchar(13) |
| ContactNo | int |

## 1.3. Staff Table Structure

Table Staff Table Structure

|  |  |
| --- | --- |
| **Field Name** | **Datatype** |
| Staff\_ID | int |
| Full\_Name | varchar(50) |
| Email | varchar(20) |
| PhoneNo | int |
| Job\_Title | varchar(50) |
| Hire\_Date | date |
| Salary | float |

## 1.4. Users Table Structure

Table Users Table Structure

|  |  |
| --- | --- |
| **Field Name** | **Datatype** |
| Username | varchar(20) |
| Password | varchar(20) |

## 1.5. Screenshot of Tables Structures

Table

Description automatically generated

Table

Description automatically generated with medium confidence

Figure Database Sructure

Table

Description automatically generatedTable

Description automatically generated

# 2. Loading Form

## 2.1. Description (How It’s Work)

The purpose of this form is to show a loading progress bar with a percentage value and transition to the next form when the loading process is completed.

The form contains a progress bar and a label to display the percentage value of the loading process. The progress bar is initialized to 0% when the form is loaded.

The form uses a timer to increment the progress bar and percentage label by 1% each time the timer ticks. The timer is set to start when the form is loaded.

When the progress bar value reaches 100%, the timer is stopped, and the current form is hidden. The next form, "Login Form", is then created and shown to the user. The Login Form is a separate form in the application that allows the user to log in and access the main features of the software.

In summary, the Loading Form is a form that provides a visual indicator of the progress of a loading process and transitions to the next form when the loading is complete.

## 2.2. Screenshot of Loading Screen

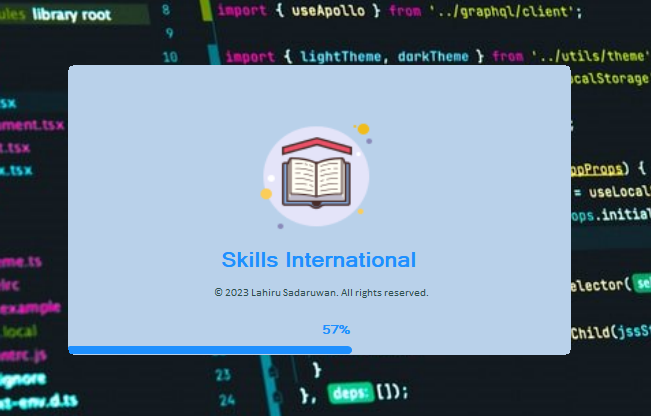


Figure Loading Screen

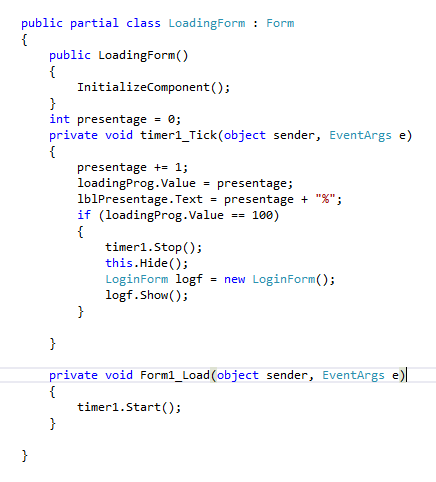


Figure Loading Form Code Snippet

# 3. SQL Connection Class

## 3.1. Description

This is a C# class called Connection. It contains a single method “GetConnection()” which returns a “SqlConnection” object, which is used to connect to a SQL Server database. The connection string for the database is specified in the “connection\_String” field of the class. This class is typically used to establish a connection to a SQL Server database.3.2. Connection Class Code Snippet

Graphical user interface, text, application

Description automatically generated

Figure Connection Class Code Snippet

# 4. Login Form

## 4.1. Description

This is a C# Windows Form application named "LoginForm". The purpose of this form is to allow users to enter their login credentials, verify them against the data stored in a database, and grant access to the software if the credentials are correct.

The form contains text boxes for the user to enter their username and password, a button to log in, a button to exit the application, and a check box to show/hide the password text. The form is loaded with the password text box showing asterisks as a security measure.

When the user clicks the "Login" button, the form connects to a SQL Server database using a connection string and executes a SELECT statement to check if the entered username and password match with the data stored in the "Users" table. If the entered credentials match a row in the table, the user is granted access to the main features of the software by hiding the login form and displaying the "HomeForm". If the entered credentials are invalid, an error message is displayed, and the user is prompted to try again.

The form also includes keyboard event handlers for the "Enter" key to move the cursor focus between the username and password fields and to simulate the "Login" button click when the user presses the "Enter" key on the password field.

## 4.2. Screenshot of Login Form

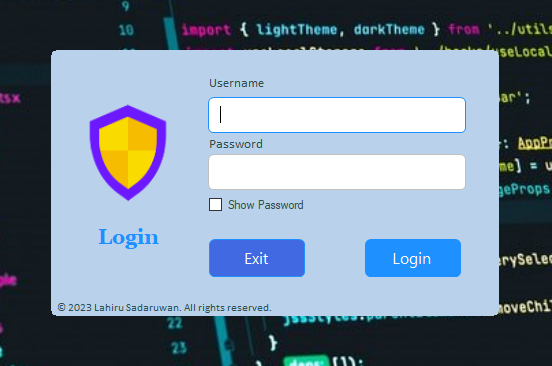


Figure Login Screen

## 4.3. Screenshot of Wrong Credential Message Box

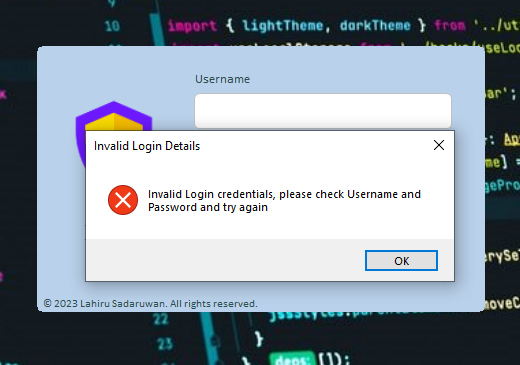


Figure Login Wrong Credentials Message

## 4.4. Screenshot of Exit Button Handle Click

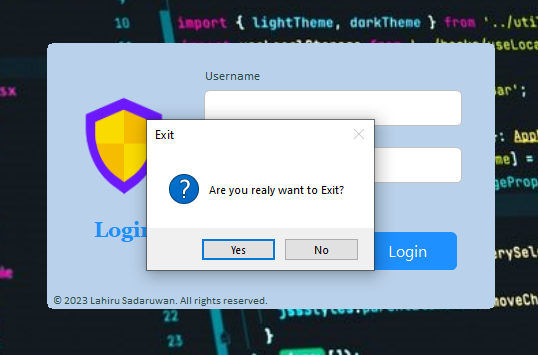


Figure Login - Exit Confirmation Message

A picture containing text

Description automatically generated

Figure Login - Import Connection Class Snippet

Graphical user interface

Description automatically generated with low confidence

Figure .Login - Exit Button Click Event Code Snippet

Graphical user interface, text, application, email

Description automatically generated

Figure Login Button Click Event Code Snippet

Graphical user interface, text, application, email

Description automatically generated

Figure Login - Password Show/Hide Code Snippet

Graphical user interface, text, application, email

Description automatically generated

Figure Login - Key Down Events Code Snippet

# 5. Home Form

## 5.1. Description

The “HomeForm” class is the main form of the "Skills International" software. It contains buttons and links for various features of the system such as student registration, managing staff, managing users, and logout. The “logoutbtn\_Click” method handles the logout process and shows a confirmation message box. The “btnStnReg\_Click”, “btnMngStaff\_Click”, and “btnMngUsers\_Click” methods handle navigation to the respective forms. There are also some click events associated with labels and images which perform the same actions as the corresponding buttons. The “HomeForm\_FormClosed” method is responsible for exiting the application when the “HomeForm” is closed.

## 5.2. Screenshot of Home Form

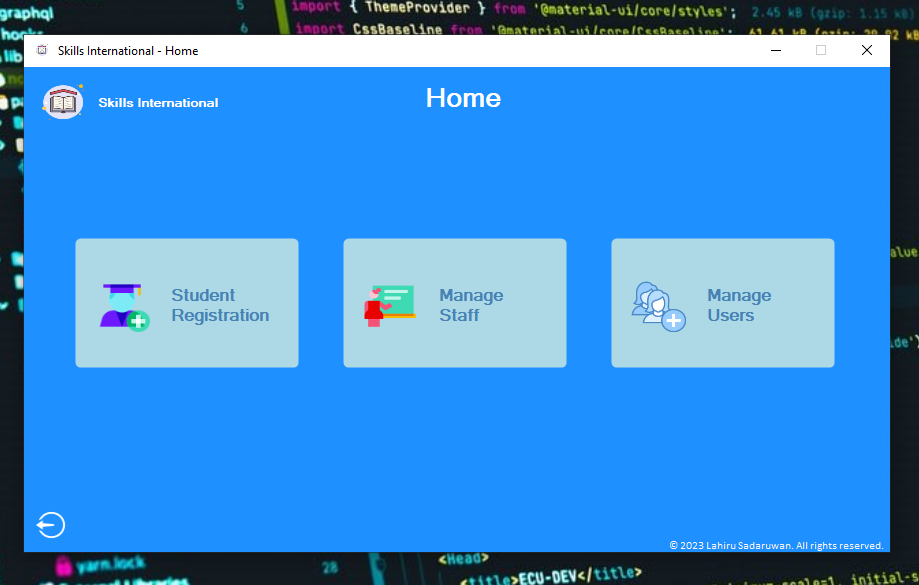


Figure Home Screen

A picture containing graphical user interface

Description automatically generated

Figure Home Form Closed Event Code Snippet

Graphical user interface, text, application

Description automatically generated

Figure Home - Navigation Button Click Events Code Snippet

Graphical user interface, application

Description automatically generated

Figure Home - Logout Confirmation Message

A picture containing text

Description automatically generated

Figure Home - Logout Event Code Snippet

# 6. Student Registration Form

## 6.1. Description

This code contains an implementation of a method to connect to the database and retrieve data to fill a DataGridView control, along with some event handlers and methods for inserting, updating and deleting data in the database.

## 6.2. Screenshot of Student Registration Form

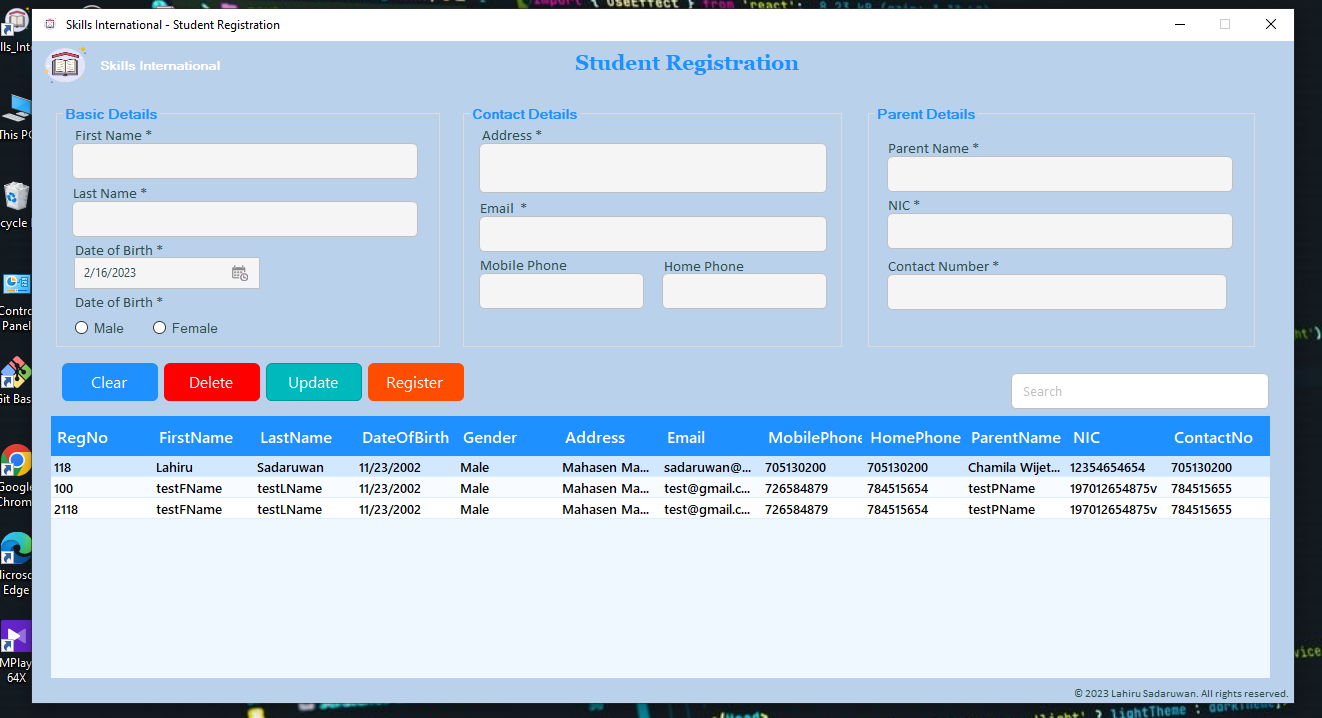


Figure Student Registration Screen

Text

Description automatically generated with medium confidence

Figure Student Registration Form Closed Event Code Snipped

Graphical user interface

Description automatically generated with medium confidence

Figure Import Connection Class Code Snippet & creates a new instance of the “SqlCommand” class Code Snippet

A picture containing chart

Description automatically generated

Figure Student Registration Form Load Event Code Snippet

Graphical user interface, text, application

Description automatically generated

Figure Search and all Data load Custom Method Code Snipped

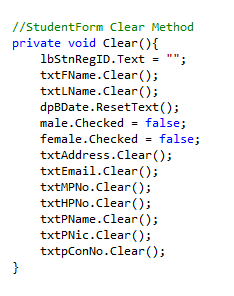


Figure Clear Custom Method Snipped

## 6.3. Registration Button Click Event Code Explanation

This is an event handler for the "Register" button click in a student registration form. When the button is clicked, the code first retrieves the data entered by the user into various textboxes and radio buttons in the form. It then checks if all the required data has been entered by the user. If any data is missing, it displays an error message. If all the required data is present, the code opens a database connection, creates an SQL command to insert the student registration data into a table, executes the command, displays a success message, clears the form for new entries, closes the connection, and reloads the data in the form. If any error occurs during this process, an error message is displayed.

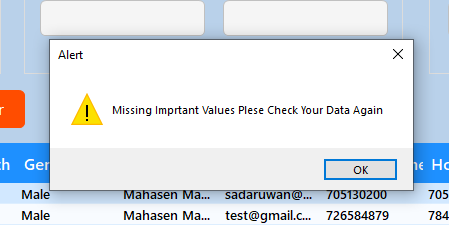


Figure Student Registration - Screenshot of Register Details Missing Warning Message Box

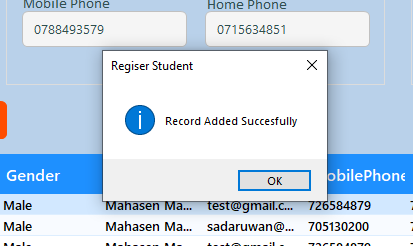


Figure Student Registration - Screenshot of Register Successful Message Box

Graphical user interface, text, application

Description automatically generated

Figure Student Registration - Register Button Click Event Code Snippet Part -1

Graphical user interface, text, application, email

Description automatically generated

Figure Student Registration - Register Button Click Event Code Snippet Part -2

## 6.4. DataGridView Click Event Description

When the user clicks on a row in the DataGridView control, the data in the selected row is used to populate the input fields of the form for editing or viewing.

Graphical user interface, application

Description automatically generated

Figure Student Registration - Screenshot of DataGridView Click Event

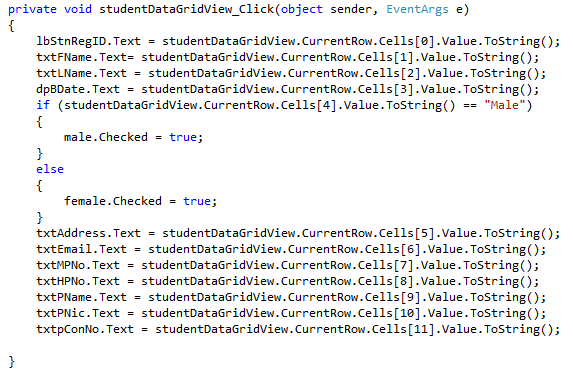


Figure Student Registration - DataGridView Click Event Code Snippet

## 6.5. Explanation of Update Button Click Event

Above code retrieves data from various input fields such as text boxes and radio buttons in a student registration form, and checks if any of the required values are missing. If any of the required values are missing, a message box with a warning message is displayed to the user. The required values include first name, last name, birth date, gender, address, email, parent name, parent NIC number, parent contact number, and student registration ID.

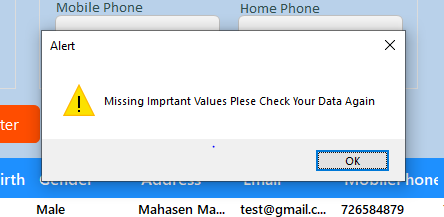


Figure Student Registration - Screenshot of Update Warning Message Box

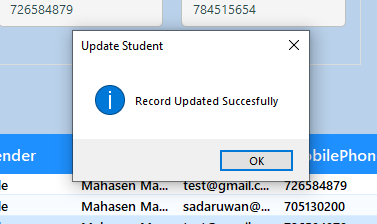


Figure Student Registration - Screenshot of Updated Successful Message Box

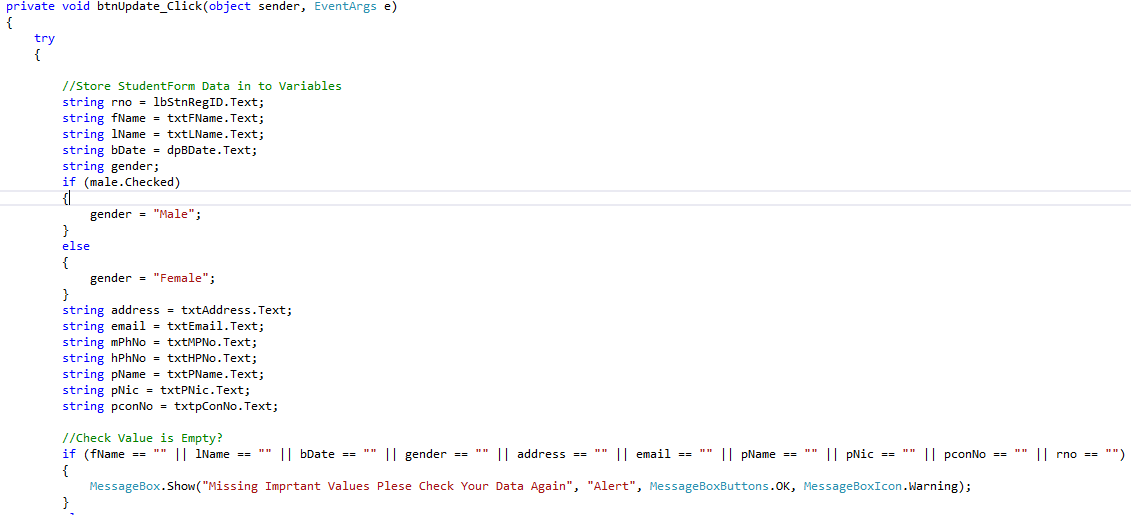


Figure Student Registration - Update Button Click Event Code Snippet Part -1



Figure Student Registration - Update Button Click Event Code Snippet Part -2

## 6.6. Explanation of Delete Button Click Event

This code is the event handler for the "Delete" button in a student registration form. When the button is clicked, a dialog box is displayed asking the user to confirm if they really want to delete the selected record. If the user clicks "Yes", the program tries to convert the value in the "RegNo" label to an integer and then uses it to construct a SQL query that deletes the record with that registration number from the database. If there is an exception while executing the query, an error message is displayed. Finally, the database connection is closed and the form's data is reloaded.

## 

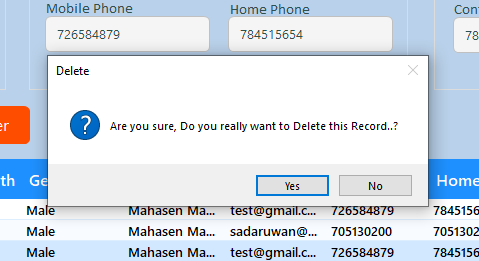


Figure Student Registration - Screenshot of Delete Confirmation Dialog Box

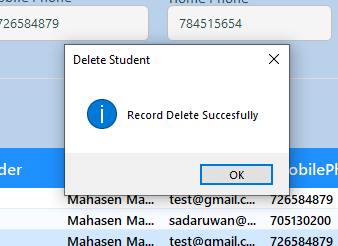


Figure Student Registration - Screenshot of Record Delete Successful Message Box

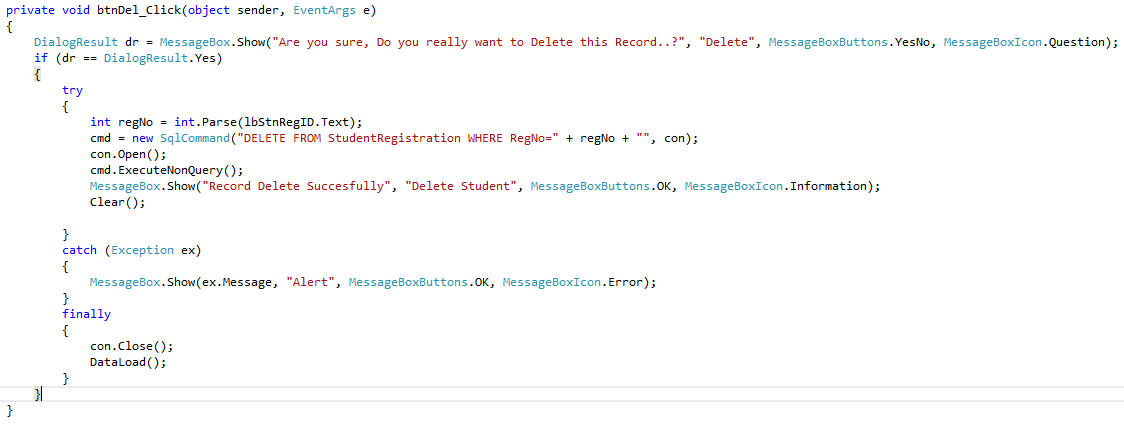


Figure Student Registration - Delete Button Click Event Code Snippet

## 6.7. Explanation of Clear Button Click Event

the "Clear" button in the student registration form. It calls the “Clear()” method, which is responsible for resetting all the input fields in the form to their default or empty state. This button provides a convenient way for users to clear the form and start again if they want to re-enter their data from scratch.

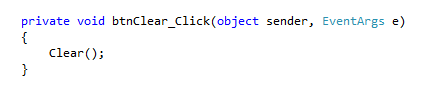


Figure Student Registration - Clear Button Code Snippet

Timeline

Description automatically generated with low confidence

Figure Screenshot of Search Student

A picture containing text

Description automatically generated

Figure Student Registration - Search Text Change Event Code Snippet

# 7. Manage Staff Form

## 7.1. Description

The application is a staff management system that allows users to add, edit, and delete staff members, as well as search for staff members in the database.

## 7.2. Screenshot of Manage Staff Form

Graphical user interface, website

Description automatically generated

Figure Screenshot of Manage Staff Form

Text

Description automatically generated with medium confidence

Figure Manage Staff Form Closed Event Code Snipped

Graphical user interface

Description automatically generated with medium confidence

Figure Import Connection Class Code Snippet & creates a new instance of the “SqlCommand” class Code Snippet

A picture containing graphical user interface

Description automatically generated

Figure Manage Staff Form Load Event Code Snippet

Graphical user interface, text, application, email

Description automatically generated

Figure Manage Staff - Search and all Data load Custom Method Code Snipped

Text, letter

Description automatically generated

Figure Manage Staff - Clear Custom Method Snipped

## 7.3. Explanation of Add Button Click Event

The purpose of the code is to add a new record to a database table named Staff.

The code starts by retrieving the values of various input fields on a form and storing them in local variables. It then checks if any of the values are empty, and displays an error message if they are.

If all the values are present, the code attempts to parse the “pNo” and “salary” variables as integers and doubles, respectively. If either of these parsing operations fails, an exception is thrown and an error message is displayed.

Assuming all the input data is valid, the code then opens a database connection and creates a new SQL command that inserts a new row into the Staff table with the values retrieved from the form. The “cmd.ExecuteNonQuery()” method is called to execute the SQL command and add the new record to the table. If the insertion is successful, a success message is displayed and the input fields on the form are cleared.

If any errors occur during the execution of the code, an exception is caught and an error message is displayed. Finally, the code closes the database connection and calls a method named “DataLoad” which presumably loads the latest data from the Staff table into the form.

## 7.4. Add Button Click Event Code Snippet

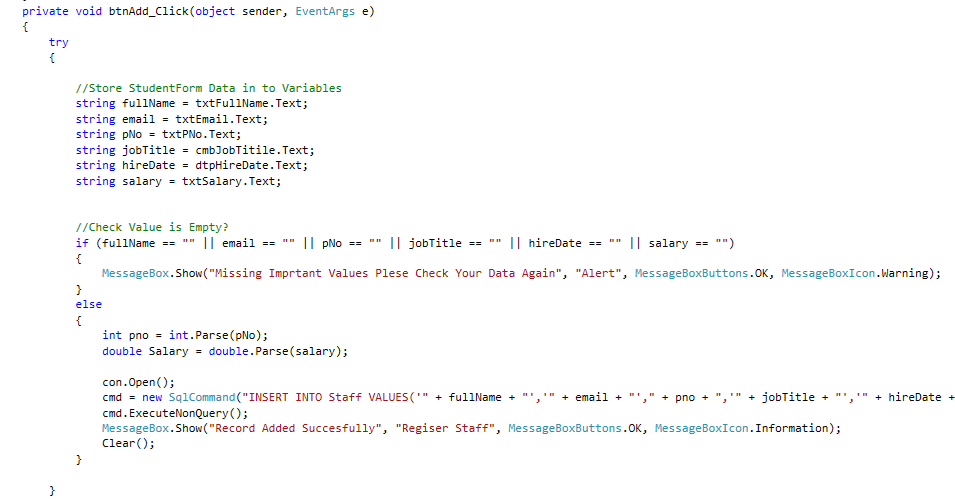


Figure Manage Staff - Add Button Click Event Code Snippet Part 1

A picture containing website

Description automatically generated

Figure Manage Staff - Add Button Click Event Code Snippet Part 2

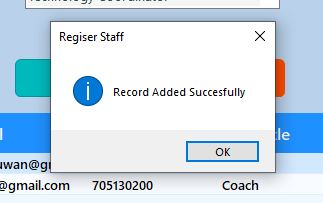


Figure Manage Staff - Screenshot of Record Added Successful Message Box

Graphical user interface, application

Description automatically generated

Figure Manage Staff - Screenshot of Record Adding Warning Message Box

## 7.5. DataGridView Click Event Description

When the user clicks on a row in the DataGridView control, the data in the selected row is used to populate the input fields of the form for editing or viewing.

Text

Description automatically generated

Figure Manage Staff - Staff DataGrid Click Event Code Snippet

Graphical user interface, website

Description automatically generated

Figure Manage Staff - Screenshot of Staff DataGrid Click Event

## 7.6. Explanation of Edit Button Click Event

When the button is clicked, the method reads the values of various input fields such as full name, email, phone number, job title, hire date, and salary, and stores them in variables. It then checks if any of the values are empty and displays a warning message if they are.

If all values are present, the method converts the phone number and salary values to their appropriate data types, and then opens a database connection. It executes a SQL UPDATE statement on the database to update the staff record corresponding to the staff ID displayed in a label named "lbStafID" with the new values.

If the update is successful, the method displays a success message and clears the input fields. If there is an error during the execution of the SQL statement, the method catches the exception, displays an error message, and closes the database connection. Finally, it reloads the data from the database into the application.

Graphical user interface, application

Description automatically generated

Figure Manage Staff - Screenshot of Editing Warning Message Box

Graphical user interface, text, application

Description automatically generated

Figure Manage Staff - Screenshot of Edited Successful Message Box

Graphical user interface, text, application, email

Description automatically generated

Figure Manage Staff - Edit Button Click Event Code Snippet Part 1

A picture containing text

Description automatically generated

Figure Manage Staff - Edit Button Click Event Code Snippet Part 2

## 7.7. Explanation of Delete Button Click Event

When the user clicks the button, a confirmation dialog is displayed asking if they are sure they want to delete the record. If the user clicks "Yes", the code retrieves the “Staff\_ID” from the “lbStafID” label, constructs a SQL DELETE statement to remove the record from the database, executes the statement, displays a message to confirm the record was deleted, and then calls the “Clear()” and “DataLoad()” methods to update the UI. If an exception occurs during the delete operation, an error message is displayed.

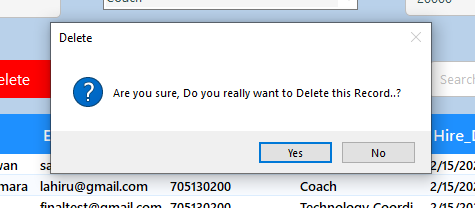


Figure Manage Staff - Screenshot of Delete Confirmation Dialog Box

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure Manage Staff - Screenshot of Record Delete Successful Message Box

Graphical user interface, text, application, email

Description automatically generated

Figure Manage Staff - Delete Button Click Event Code Snippet

## 7.8. Explanation of Clear Button Click Event

The "Clear" button in the Manage Staff form. It calls the “Clear()” method, which is responsible for resetting all the input fields in the form to their default or empty state. This button provides a convenient way for users to clear the form and start again if they want to re-enter their data from scratch.

Graphical user interface

Description automatically generated with medium confidence

Figure Manage Staff - Clear Button Code Snippet

Graphical user interface

Description automatically generated

Figure Manage Staff - Screenshot of Search Student

A picture containing text

Description automatically generated

Figure Manage Staff - Search Text Change Event Code Snippet

# 8. Users Form

## 8.1. Description

This Form for managing user records using a SQL Server database. It allows users to add, edit, and delete user records, as well as view a list of all current users. The application contains a form with input fields for username, password, and confirm password, and buttons for adding, editing, deleting, and clearing user records. The application connects to a SQL Server database to store and retrieve user records.

## 8.2. Screenshot of Users Form

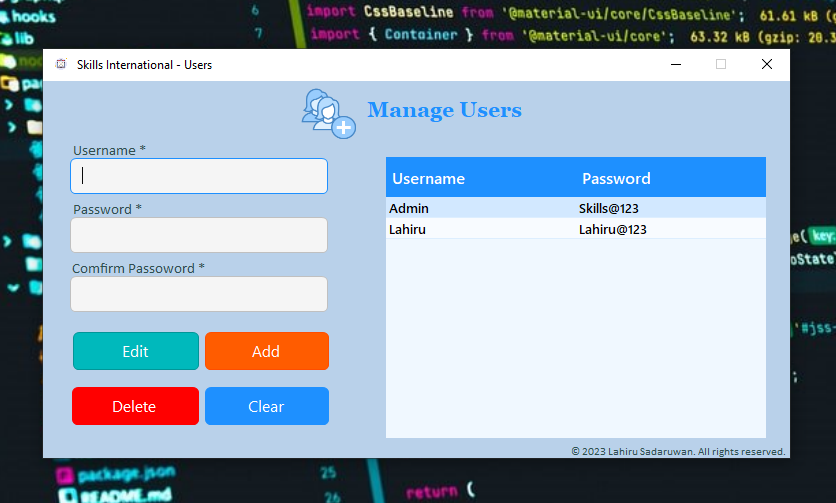


Figure Manage Users Form Screen

## 8.3.

Text

Description automatically generated

Figure Users Form Closed Event Code Snipped

Graphical user interface

Description automatically generated with medium confidence

Figure Import Connection Class Code Snippet & creates a new instance of the “SqlCommand” class Code Snippet

Graphical user interface, application

Description automatically generated

Figure Manage Users - Users Form Load Event Code Snippet

Text

Description automatically generated

Figure Manage Users - Data load Custom Method Code Snipped

Text, letter

Description automatically generated

Figure Manage Users - Clear Custom Method Snipped

## 8.3. Explanation of Add Button Click Event

When the user clicks the button, the code first retrieves the username and password values entered by the user. It then checks if the two password fields match, and if they do, stores the password value. If the two password fields do not match, a message box is displayed to inform the user. The code then checks if the username and password fields are not empty. If they are, a warning message box is displayed. If the fields are not empty, the code opens a connection to the database and creates a SQL command to insert the username and password values into a "Users" table. The SQL command is executed and a success message box is displayed. Finally, the code clears the form and reloads the data. If any error occurs during this process, an error message box is displayed.



Figure Manage Users - Add Button Click Event Code Snippet Part 1

Graphical user interface

Description automatically generated with medium confidence

Figure Manage Users - Add Button Click Event Code Snippet Part 2

Graphical user interface, text, application

Description automatically generated

Figure Manage Users - Screenshot of Record Added Successful Message Box

Graphical user interface, text, application

Description automatically generated

Figure Manage Users - Screenshot of Record Adding Warning Message Box

## 8.4. DataGridView Click Event Description

When the user clicks on a row in the DataGridView control, the data in the selected row is used to populate the input fields of the form for editing or viewing.

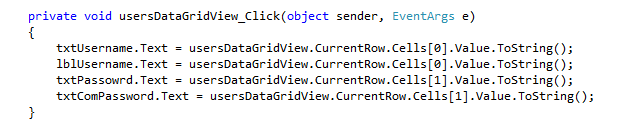


Figure Manage Users - Users DataGrid Click Event Code Snippet

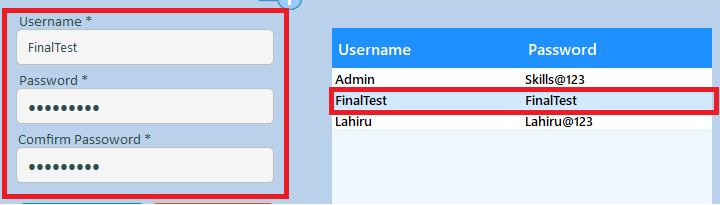


Figure Manage Users - Screenshot of Users DataGrid Click Event

## 8.5. Explanation of Edit Button Click Event

When the button is clicked, the function will attempt to update the user's data in the database.

First, the code checks if the password and confirm password fields match. If they do, the password is stored in a variable called "password." If they don't match, a message box is displayed to the user indicating the mismatch.

Then, the code checks if the "Username" and "Password" fields are empty. If either field is empty, a warning message box is displayed to the user indicating that the important values are missing.

If both fields have values, the function establishes a connection to the database and updates the user's data using the "UPDATE" SQL statement. The command object is used to execute the statement, and if the update is successful, a message box is displayed indicating that the record has been updated. Finally, the "Clear" function is called to clear the form, and the "DataLoad" function is called to refresh the data displayed in the form. If any errors occur during the execution of the code, an error message box is displayed to the user.

Graphical user interface, text, application

Description automatically generated

Figure Manage Users - Screenshot of Editing Warning Message Box

Graphical user interface, text, application

Description automatically generated

Figure Manage Users - Screenshot of Edited Successful Message Box

A picture containing text

Description automatically generated

Figure Manage Users - Edit Button Click Event Code Snippet Part 1

A picture containing text

Description automatically generated

Figure Manage Users - Edit Button Click Event Code Snippet Part 2

## 8.6. Explanation of Delete Button Click Event

When the button is clicked, a message box will appear asking the user to confirm if they really want to delete the selected record. If the user selects "Yes", the code will try to delete the record from the database.

First, the username of the selected record is stored in a variable. Then, a SQL command is created to delete the record with that username from the database. The command is executed, and a message box is displayed to inform the user that the record has been deleted. If an exception occurs while executing the command, an error message box will appear.

Finally, the database connection is closed and the “DataLoad” function is called to refresh the data displayed in the user interface.

## 

Graphical user interface, text, application

Description automatically generated

Figure Manage Users - Screenshot of Delete Confirmation Dialog Box

Graphical user interface, text, application

Description automatically generated

Figure Manage Users - Screenshot of Record Delete Successful Message Box

Graphical user interface, text, application

Description automatically generated

Figure Manage Users - Delete Button Click Event Code Snippet

## 8.7. Explanation of Clear Button Click Event

The "Clear" button in the Users form. It calls the “Clear()” method, which is responsible for resetting all the input fields in the form to their default or empty state. This button provides a convenient way for users to clear the form and start again if they want to re-enter their data from scratch.

Graphical user interface

Description automatically generated with medium confidence

Figure Manage Users - Clear Button Code Snippet