ABC Education Institute Student Management System

Name- D.L.Chandeepa Janith Peiris

Registration Number – NUG/A-009260

Batch No – 324

Acknowledgement

I would like to express my special thanks of gratitude to my teacher Miss.Ravini who gave me this golden opportunity to do this project of “ABC education institute student management system”. I came to know about so many new things from this project and I am really thankful to them. Secondly, I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

Content

1.0 Introduction

Student Management System is a software which is helpful for students as well as the administration authorities. This project is developed mainly to administrate the student records rather easily and comfortably than the existing system. This software of student management system can be used in various educational institutes across the globe and simplifies working of institutes.

a. Company overview

Institute name: ABC Educational Institute

Owner: Mr. J.H. Perera

Location: Nugegoda, SriLanka

b. Existing system with drawbacks

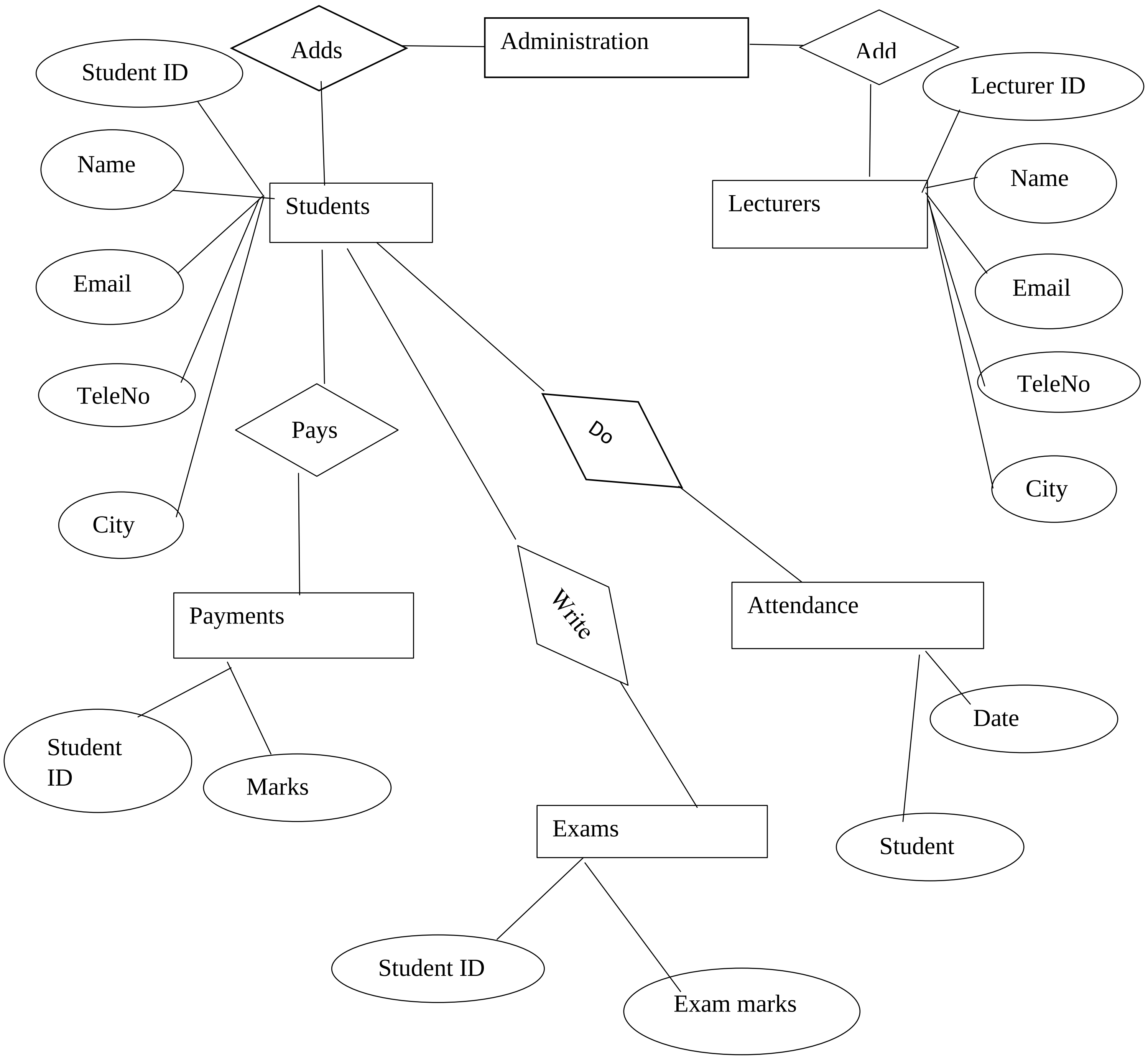
In the current system they need to keep a number of records related to the student, which is very annoying. And the administration has to enter the details of the student, enter the marks of the student, mark the attendance of students all by themselves manually. So, these kinds of things are very time consuming and costly for the institute. So computerized system will be much easier and more efficient for the institute.

c. Proposed System

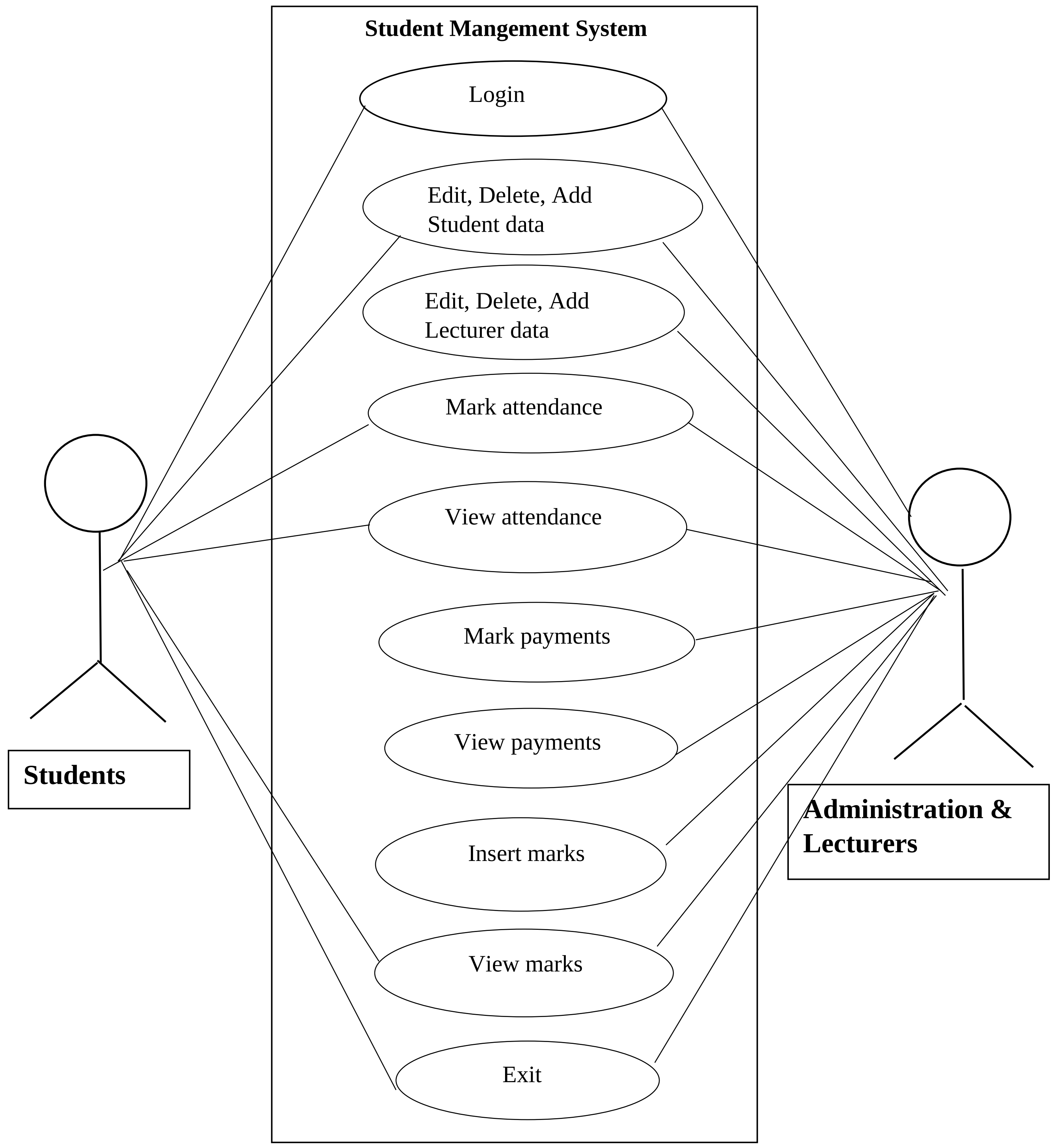
In the proposed system we have the provision for adding the details of the students by themselves. Another advantage of the system is that it is very easy to edit the details of the students and delete a student when it found unnecessary. The marks of the students are added in the database and students can view marks whenever they want. Marking attendance of students can be done by students themselves. Student payments are instantly marked when they done the payments. So like that our new system can be very effective to this institute.

Diagrams

a) ER Diagram

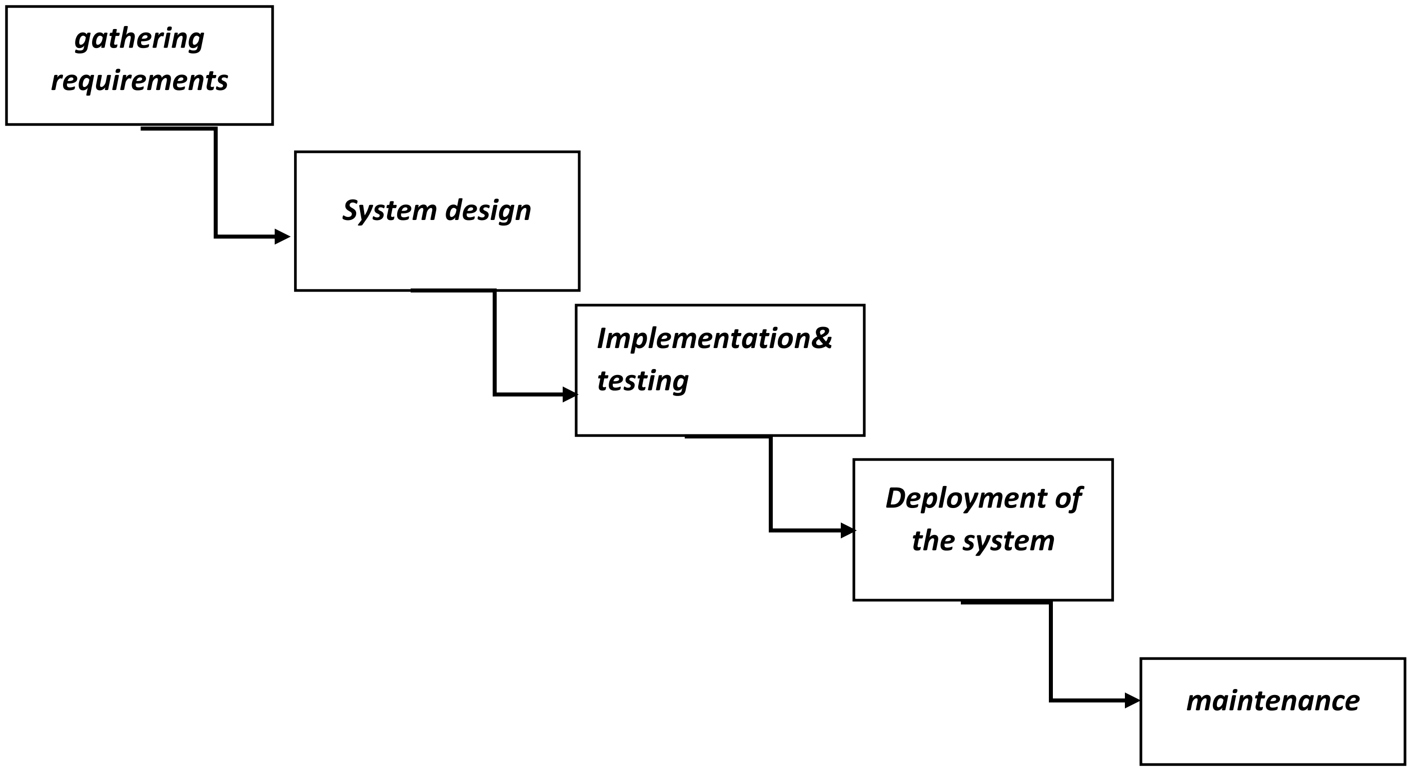


b) Use Case Diagram



3.0 Software development model

- We used waterfall model in order to develop the software.



A. Gathering Requirements

As to begin the development of the software we have to gather requirements. we gathered data and information from several methods. Analyzing documents of the institute, Interview students and administration, surveys and questionaries, prototyping are the methods we used to gather information. So, we could get thorough understanding what kind of a system that they are expecting us to build.

B. System design

After gathering the information about the system, the next step is to design the system. In order to design the system, we are using the "VISUAL STUDIO.NET" software as the user interface “C#" as the programming language and a local database create inside visual studio.

C. implementation & testing

Implementation is the stage where the theoretical design is turned into a working system. Successful completion of the implementation phrase should comprise system installation and training of end users on the system. So, after installing system to the ABC institute's computer network, we have to train the administration and working staff how to work with this new implemented system.

Testing is the process of evaluating a system or its components with the intent to find whether it satisfies the specified requirements or not. So, from this we can find out what are the errors, missing requirements etc.

D. Deployment of the system

In this phrase, the software is deployed into a live environment (client's server) in order to test its performance. Once the software is deployed, it becomes available to end users which are administration of the institute. This is the stage the institute's administration and working staff start using the new system.

E. Maintenance

After the deployment phrase the next step is to provide support and maintenance for the software, making sure it runs smoothly. If the client and end-users come across errors, defects, bugs etc. during use we have to modify the system in order to correct faults and improve performance of the system.

4.0 Functional and Non-Functional Requirements

Functional Requirements

1. Student Information management
2. Lecturer Information management
3. Student Exam marks management
4. Student attendance management
5. Student payment management

Non-Functional Requirements

1.User friendly environment

2.Mainatainability

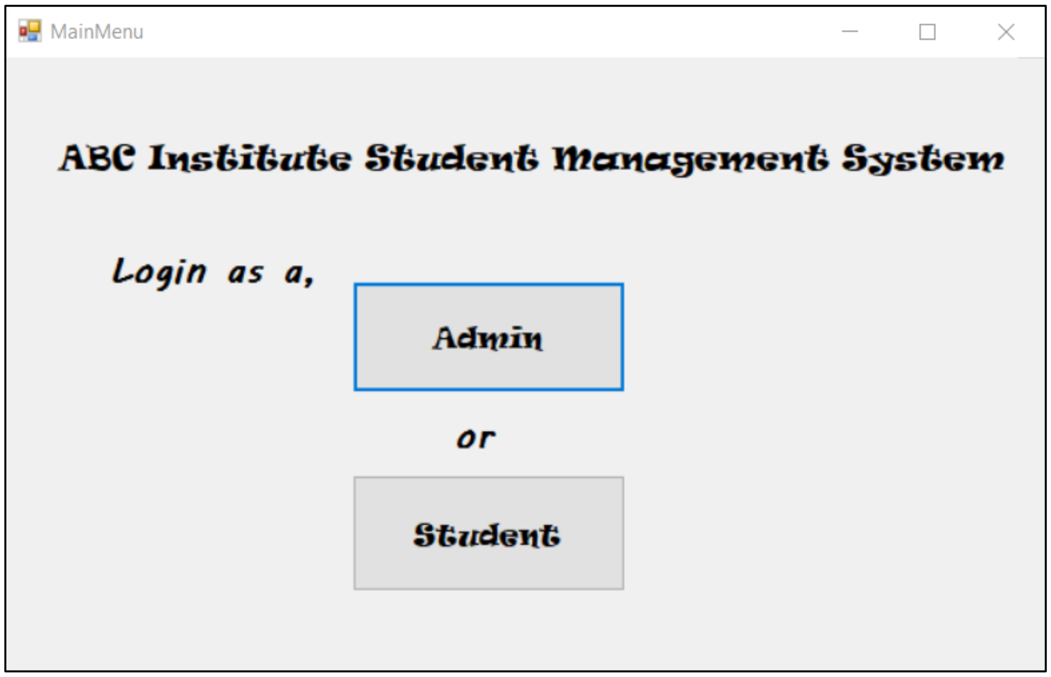
3.Accessibility

4.Durability

Interface Designing & Coding

When the user open the application, you will get “Main Menu” form.

Main Menu will view as below.



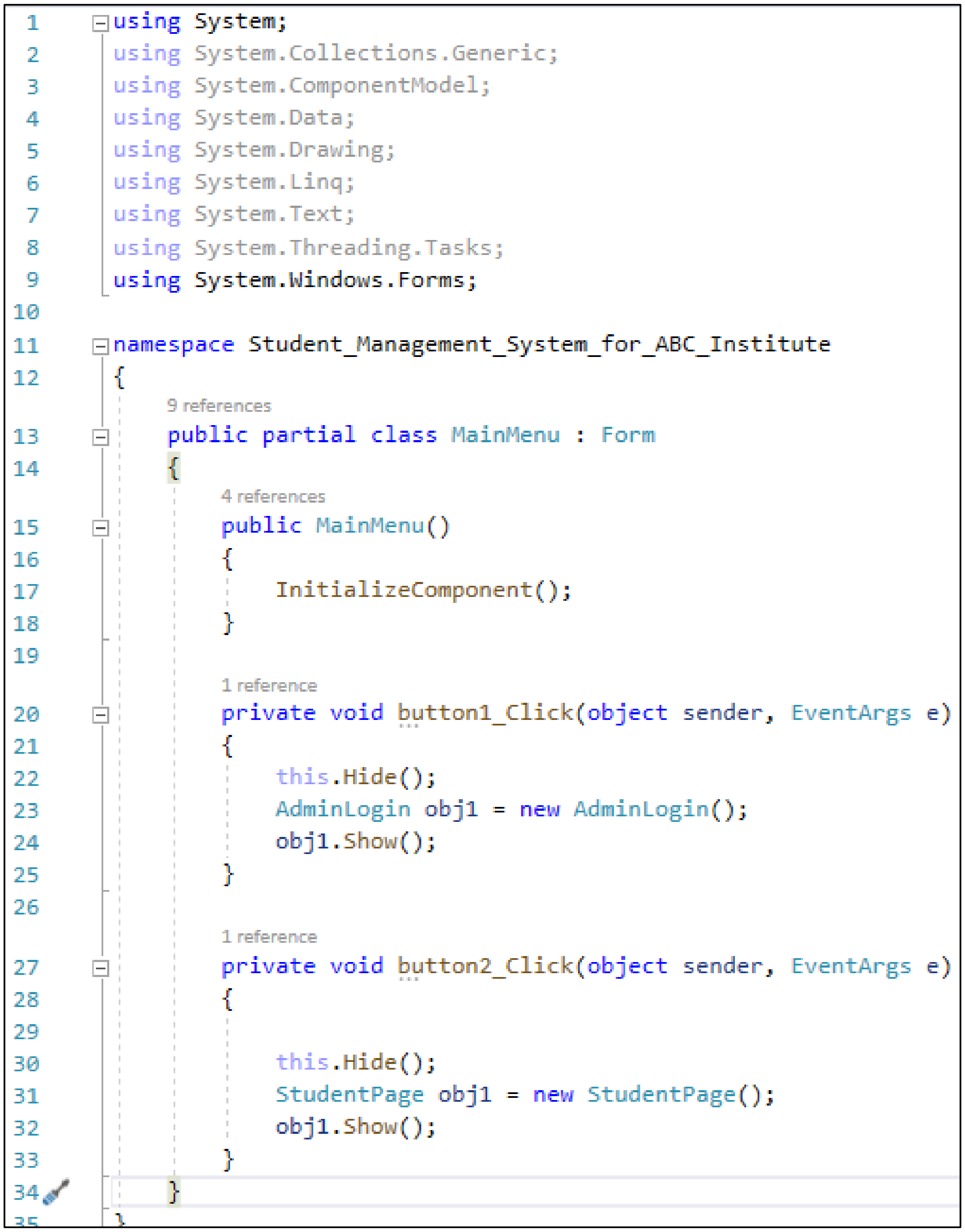
Here the user can login as,

1) Admin

or

2) Student.

Main menu coding



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Student_Management_System_for_ABC_Institute
{
    9 references
    public partial class MainMenu : Form
    {
        4 references
        public MainMenu()
        {
            InitializeComponent();
        }

        1 reference
        private void button1_Click(object sender, EventArgs e)
        {
            this.Hide();
            AdminLogin obj1 = new AdminLogin();
            obj1.Show();
        }

        1 reference
        private void button2_Click(object sender, EventArgs e)
        {

            this.Hide();
            StudentPage obj1 = new StudentPage();
            obj1.Show();
        }
    }
```

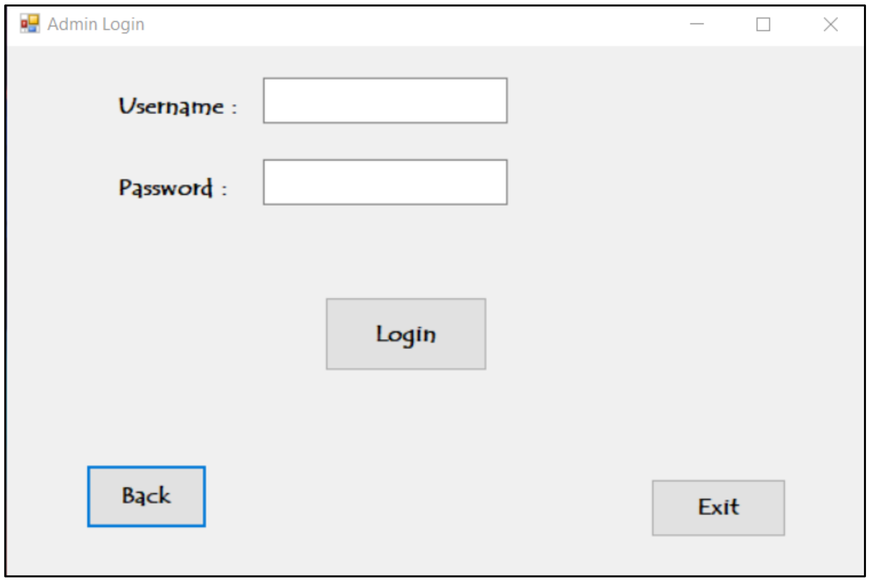
1. <u>Admin Button</u>

After clicking Admin button, user automatically go to Admin login form.

Admin Login Page

By clicking the admin button, user can access to the admin Login Page.

Admin Login Page will view as below.

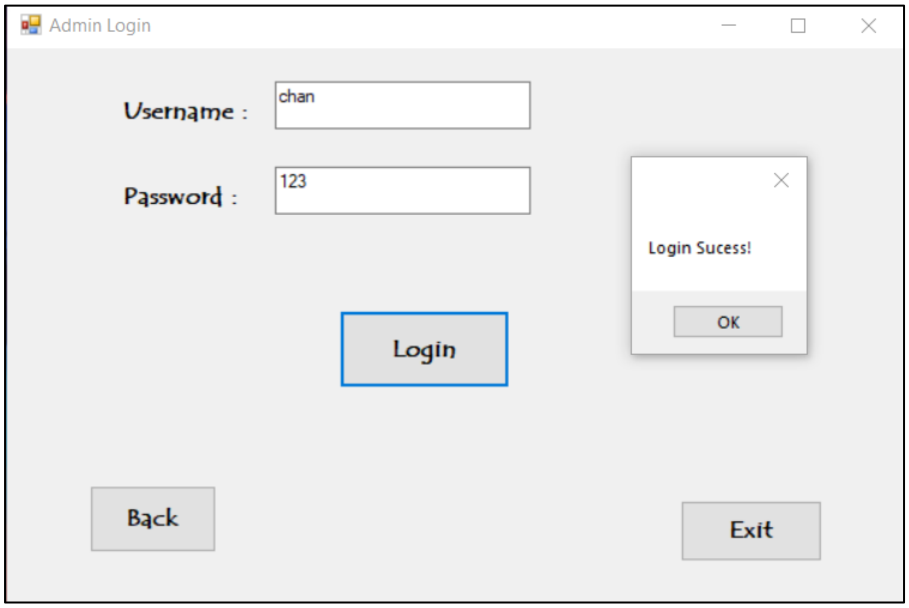


You can see two textboxes in Admin Login Page. As to Login to Admin Page user will have to enter correct username and password.

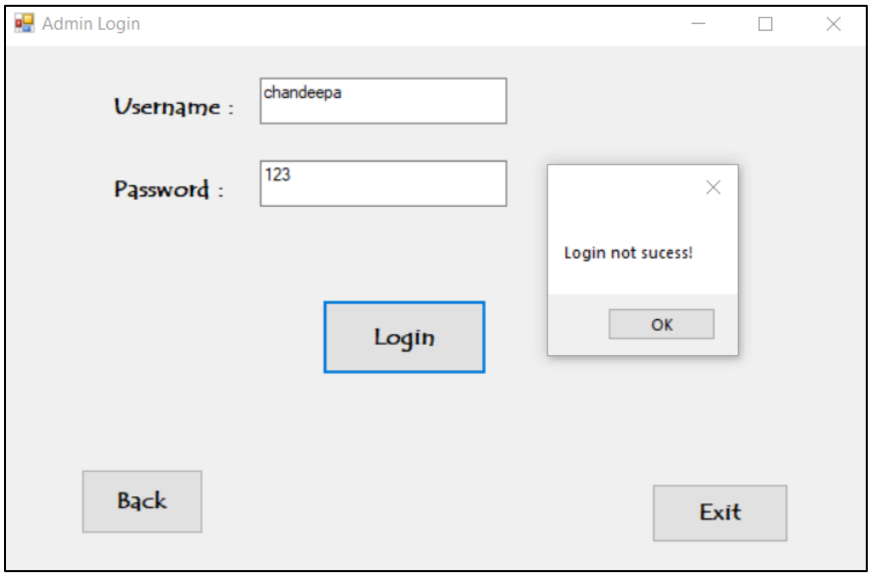
Username= "chan"

Password= "123"

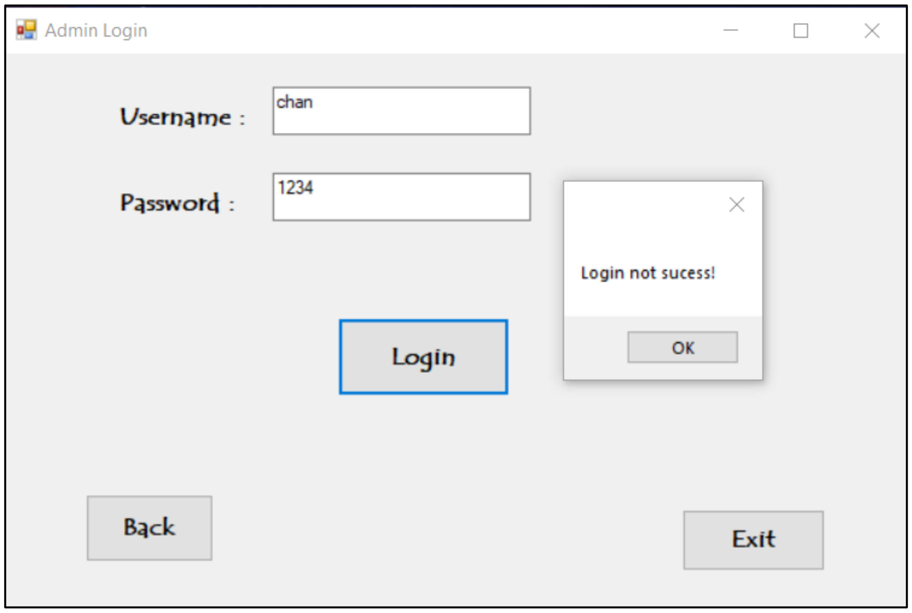
- If the user enter correct username & password user can log into the Admin Page. Then after clicking Login button, you can get a message box that shows the login is successful. And now user can access to the Admin Page.



- If the entered username is incorrect user cannot log into the Admin Page. When you click Login button you will get a message box that show the login is not successful.

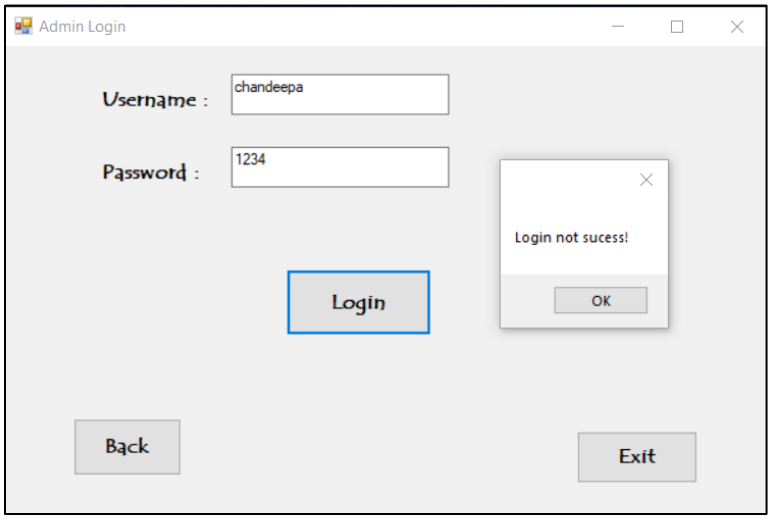


- If the entered Password is incorrect user cannot log into the Admin Page. When you click Login button you will get a message box that shows login is not successful.



Login not sucess!

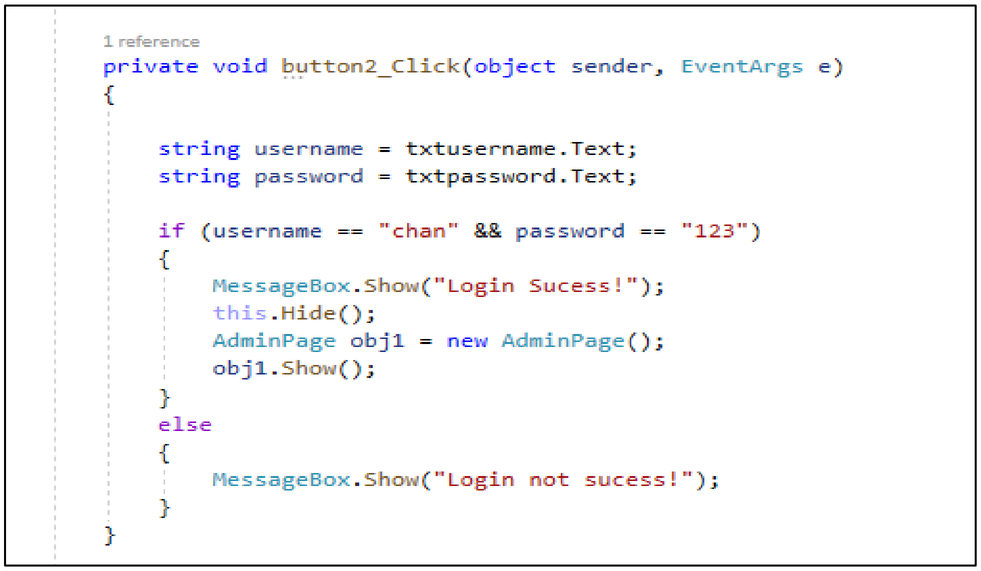
- If the entered Username and Password both are incorrect user cannot log into the Admin Page. When you click Login button you will get a message box that shows login is not successful.



Login not sucess!

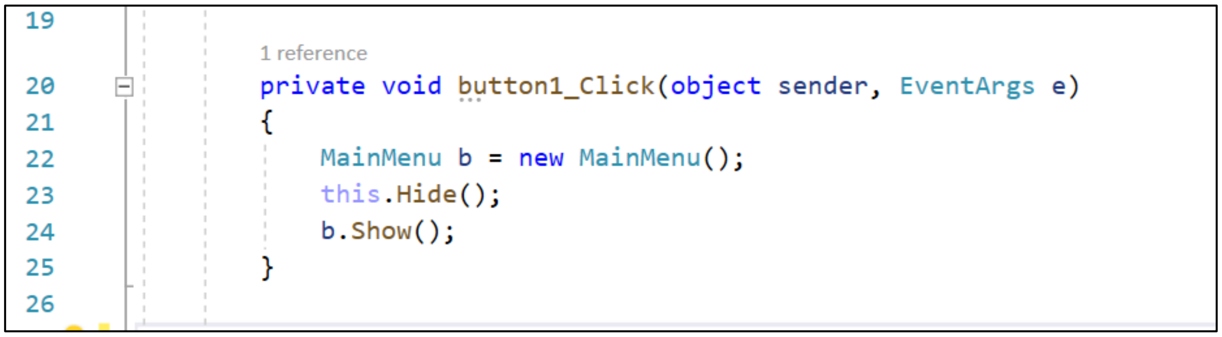
So, in order to have the access to the Admin page the user must enter the correct username & password properly. Otherwise, the user cannot access to the Admin page.

Login button coding



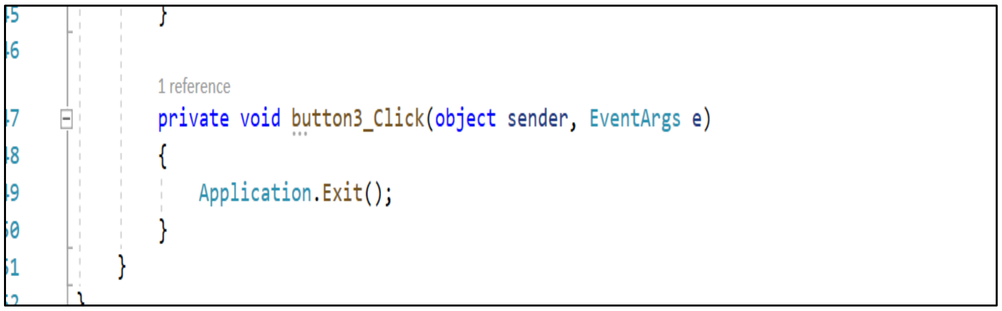
```
1 reference
private void button2_Click(object sender, EventArgs e)
{

    string username = txtusername.Text;
    string password = txtpassword.Text;

    if (username == "chan" && password == "123")
    {
        MessageBox.Show("Login Sucess!");
        this.Hide();
        AdminPage obj1 = new AdminPage();
        obj1.Show();
    }
    else
    {
        MessageBox.Show("Login not sucess!");
    }
}
```

Back button coding



```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    MainMenu b = new MainMenu();
    this.Hide();
    b.Show();
}
```

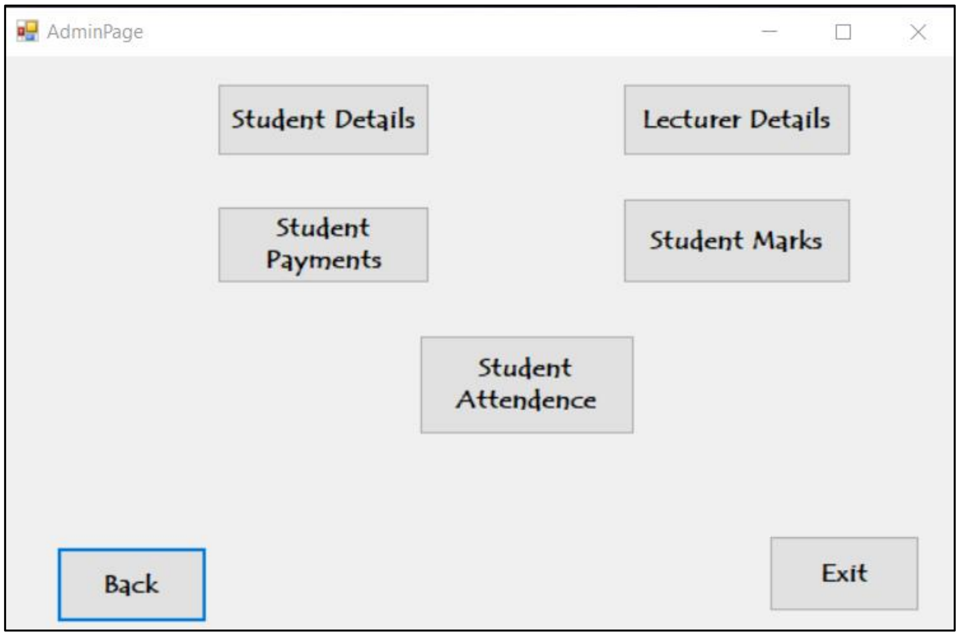
Exit button coding



```
    }

    1 reference
    private void button3_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }
}
```

After entering correct username & password user can access into the Admin Page.

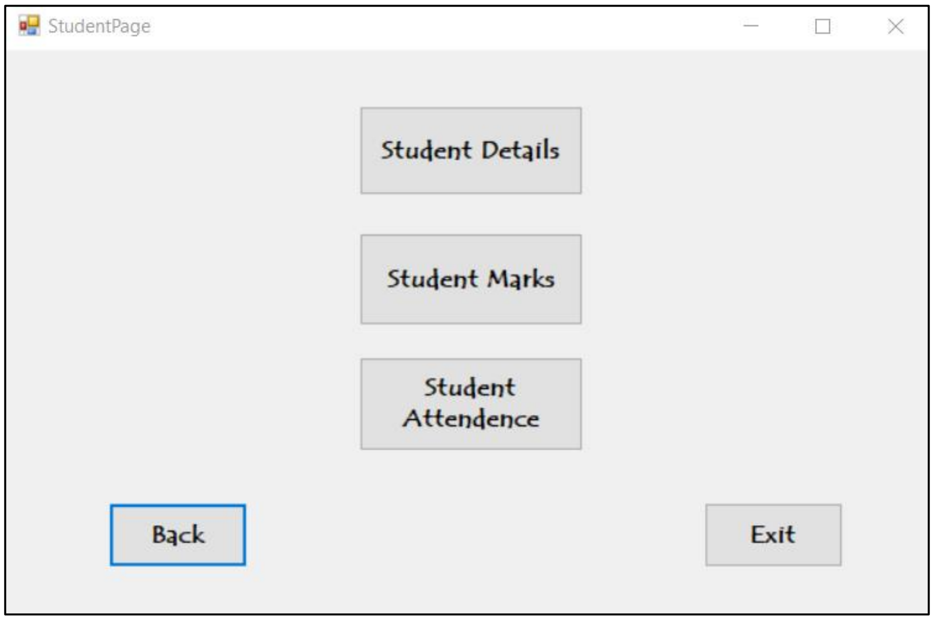
Admin Page will view as below.



2. <u>Student Button</u>

After clicking the student button in Main Menu user can access into the Student Page.

Student page will view as below.



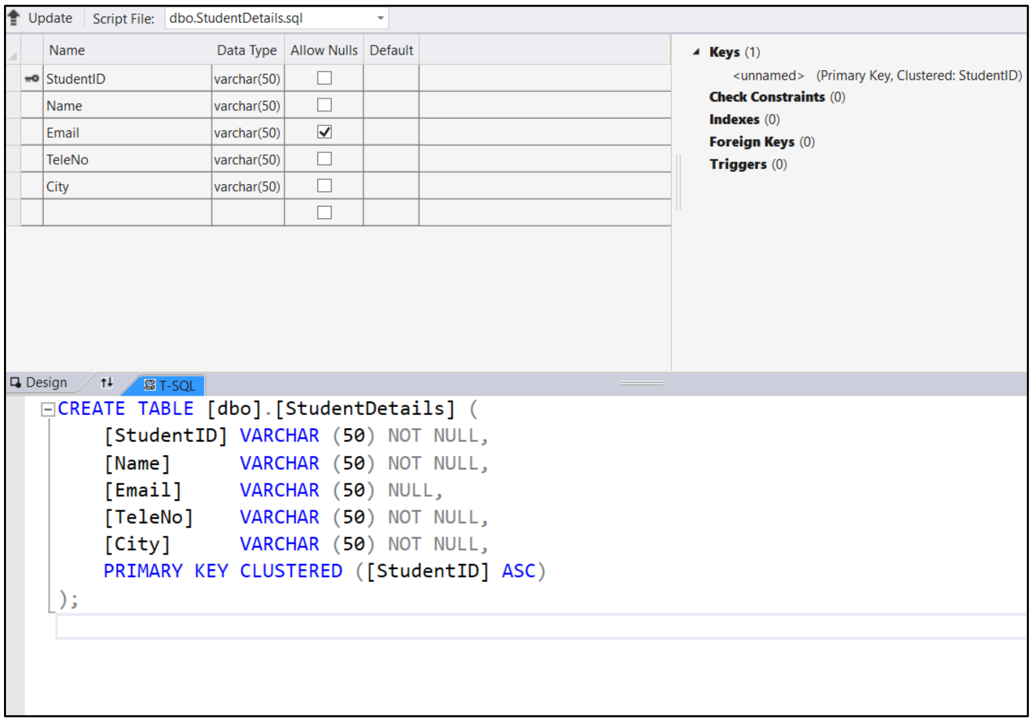
Before discussing about Admin page and Student page, let's look at the database used in this system

In this student management system, we used one database which consists of 5 tables.

Those tables are,

1) Student Details table
2) Student Marks table
3) Student Payments table
4) Student Attendance table
5) Lecturer Details table

Student Details table designing & coding

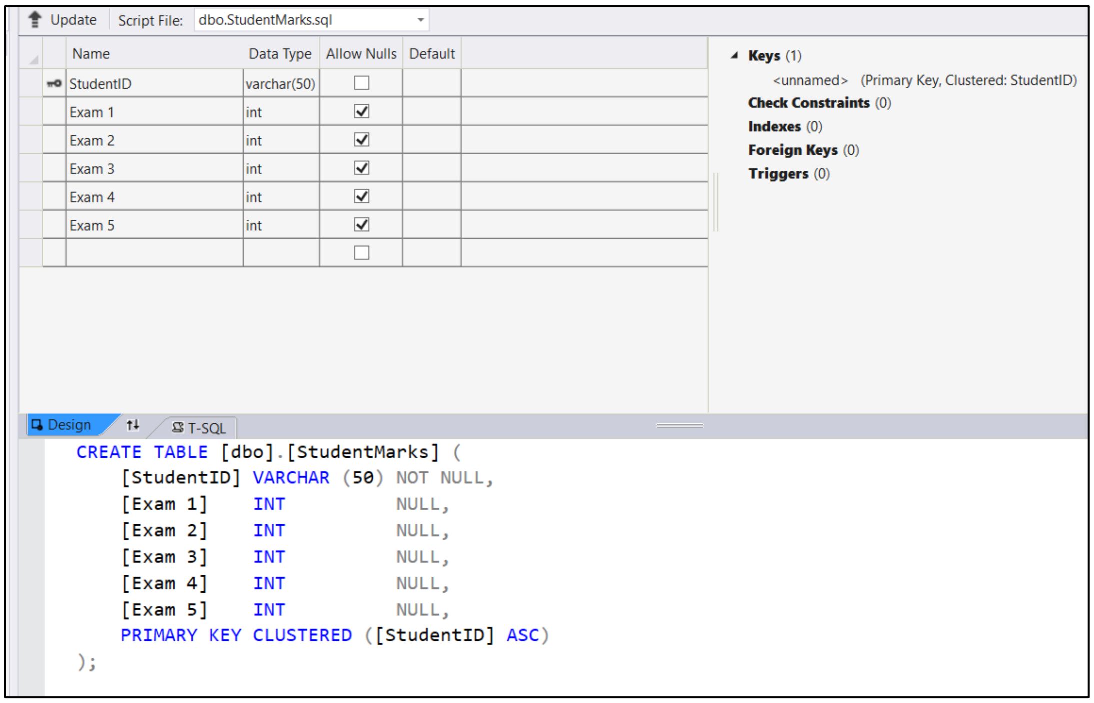


Name	Data Type	Allow Nulls	Default
StudentID	varchar(50)	☐	
Name	varchar(50)	☐	
Email	varchar(50)	✓	
TeleNo	varchar(50)	☐	
City	varchar(50)	☐	
		☐	

Keys (1)
<unnamed> (Primary Key, Clustered: StudentID)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

```
CREATE TABLE [dbo].[StudentDetails] (
    [StudentID] VARCHAR (50) NOT NULL,
    [Name]      VARCHAR (50) NOT NULL,
    [Email]     VARCHAR (50) NULL,
    [TeleNo]    VARCHAR (50) NOT NULL,
    [City]      VARCHAR (50) NOT NULL,
    PRIMARY KEY CLUSTERED ([StudentID] ASC)
);
```

Student Marks table designing & coding

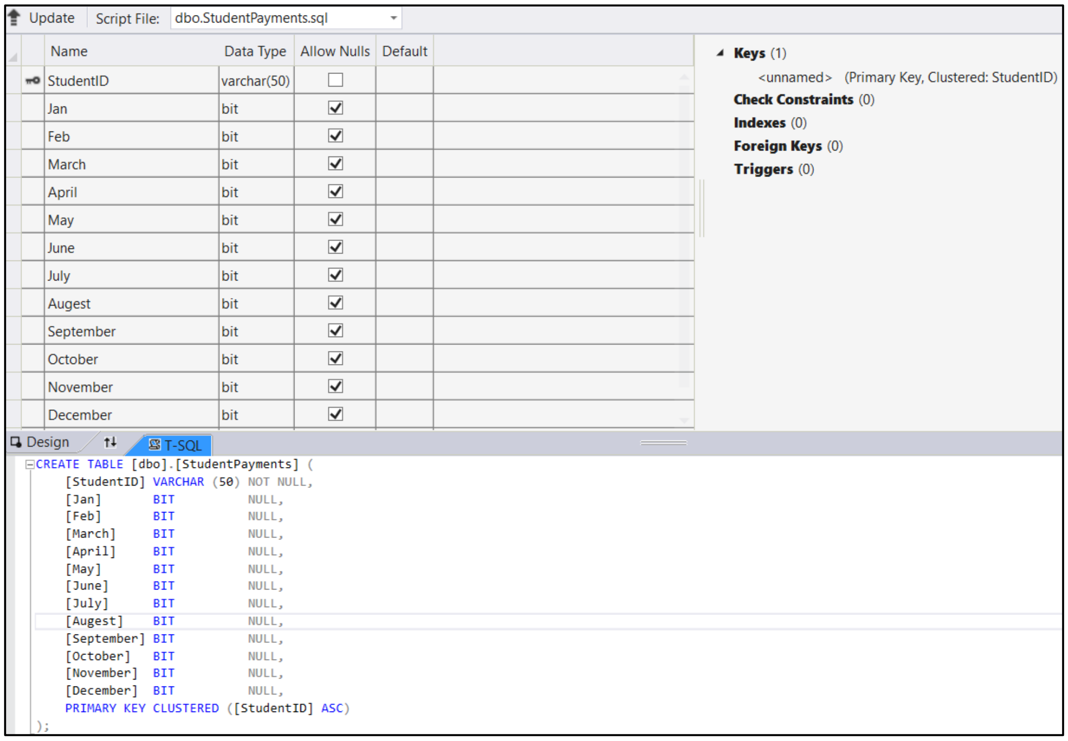


Name	Data Type	Allow Nulls	Default
StudentID	varchar(50)	☐	
Exam 1	int	✓	
Exam 2	int	✓	
Exam 3	int	✓	
Exam 4	int	✓	
Exam 5	int	✓	
		☐	

Keys (1)
<unnamed> (Primary Key, Clustered: StudentID)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

```
CREATE TABLE [dbo].[StudentMarks] (
    [StudentID] VARCHAR (50) NOT NULL,
    [Exam 1]    INT          NULL,
    [Exam 2]    INT          NULL,
    [Exam 3]    INT          NULL,
    [Exam 4]    INT          NULL,
    [Exam 5]    INT          NULL,
    PRIMARY KEY CLUSTERED ([StudentID] ASC)
);
```

Student Payments table designing & coding

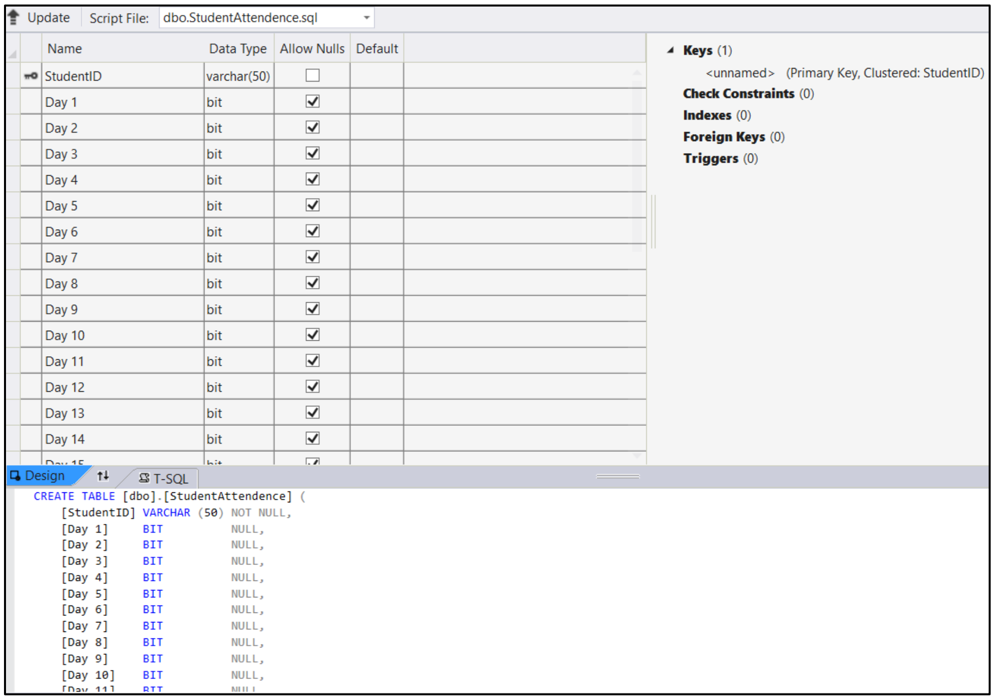


Name	Data Type	Allow Nulls	Default
StudentID	varchar(50)	☐	
Jan	bit	✓	
Feb	bit	✓	
March	bit	✓	
April	bit	✓	
May	bit	✓	
June	bit	✓	
July	bit	✓	
Augest	bit	✓	
September	bit	✓	
October	bit	✓	
November	bit	✓	
December	bit	✓	

Keys (1)
<unnamed> (Primary Key, Clustered: StudentID)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

```
CREATE TABLE [dbo].[StudentPayments] (
    [StudentID] VARCHAR (50) NOT NULL,
    [Jan]       BIT          NULL,
    [Feb]       BIT          NULL,
    [March]     BIT          NULL,
    [April]     BIT          NULL,
    [May]       BIT          NULL,
    [June]      BIT          NULL,
    [July]      BIT          NULL,
    [Augest]    BIT          NULL,
    [September] BIT          NULL,
    [October]   BIT          NULL,
    [November]  BIT          NULL,
    [December]  BIT          NULL,
    PRIMARY KEY CLUSTERED ([StudentID] ASC)
);
```

Student Attendance table designing & coding

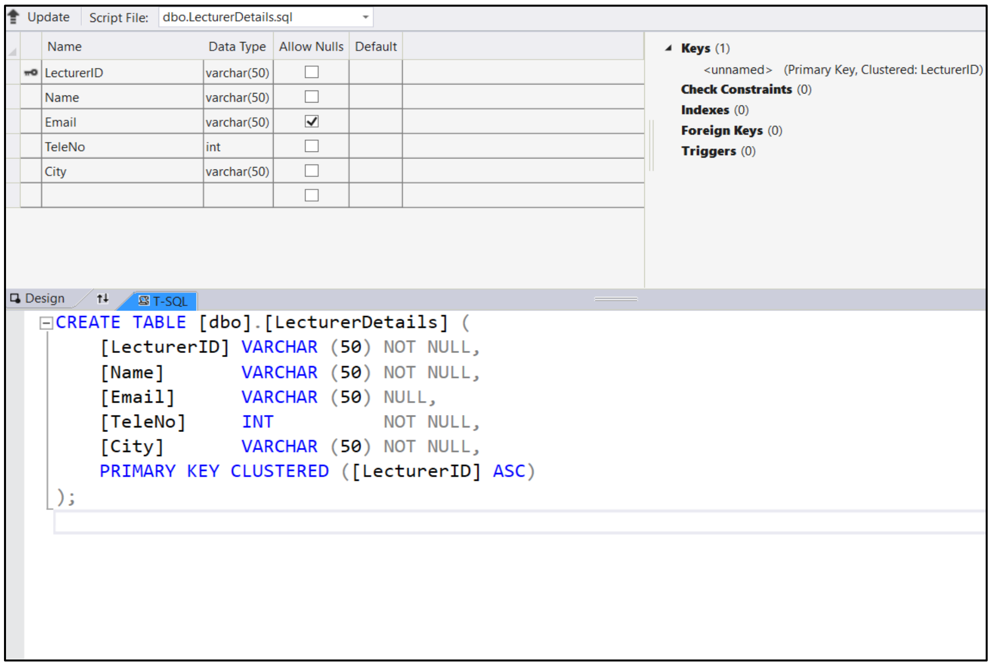


Name	Data Type	Allow Nulls	Default
StudentID	varchar(50)	☐	
Day 1	bit	✓	
Day 2	bit	✓	
Day 3	bit	✓	
Day 4	bit	✓	
Day 5	bit	✓	
Day 6	bit	✓	
Day 7	bit	✓	
Day 8	bit	✓	
Day 9	bit	✓	
Day 10	bit	✓	
Day 11	bit	✓	
Day 12	bit	✓	
Day 13	bit	✓	
Day 14	bit	✓	

Keys (1)
<unnamed> (Primary Key, Clustered: StudentID)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

```
CREATE TABLE [dbo].[StudentAttendence] (
    [StudentID] VARCHAR (50) NOT NULL,
    [Day 1]     BIT          NULL,
    [Day 2]     BIT          NULL,
    [Day 3]     BIT          NULL,
    [Day 4]     BIT          NULL,
    [Day 5]     BIT          NULL,
    [Day 6]     BIT          NULL,
    [Day 7]     BIT          NULL,
    [Day 8]     BIT          NULL,
    [Day 9]     BIT          NULL,
    [Day 10]    BIT          NULL,
```

Lecturer Details table designing & coding



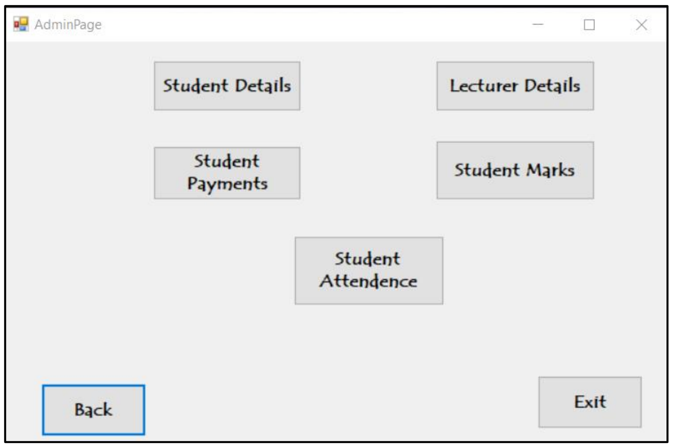
Name	Data Type	Allow Nulls	Default
LecturerID	varchar(50)	☐	
Name	varchar(50)	☐	
Email	varchar(50)	✓	
TeleNo	int	☐	
City	varchar(50)	☐	
		☐	

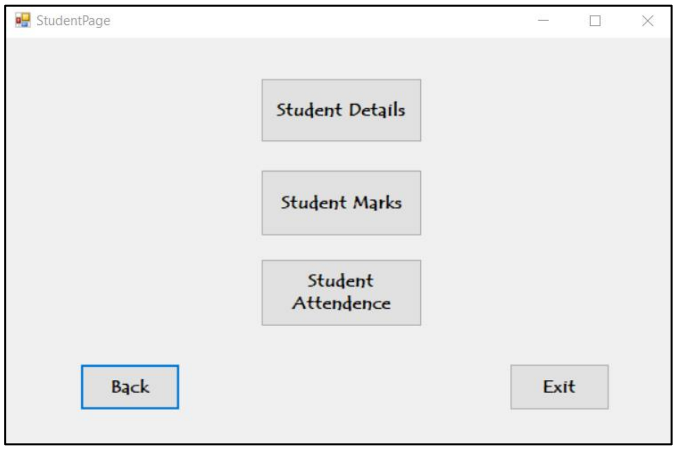
Keys (1)
<unnamed> (Primary Key, Clustered: LecturerID)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

```
CREATE TABLE [dbo].[LecturerDetails] (
    [LecturerID] VARCHAR (50) NOT NULL,
    [Name]       VARCHAR (50) NOT NULL,
    [Email]      VARCHAR (50) NULL,
    [TeleNo]     INT          NOT NULL,
    [City]       VARCHAR (50) NOT NULL,
    PRIMARY KEY CLUSTERED ([LecturerID] ASC)
);
```

So now let's discuss about Admin Page and Student Page.

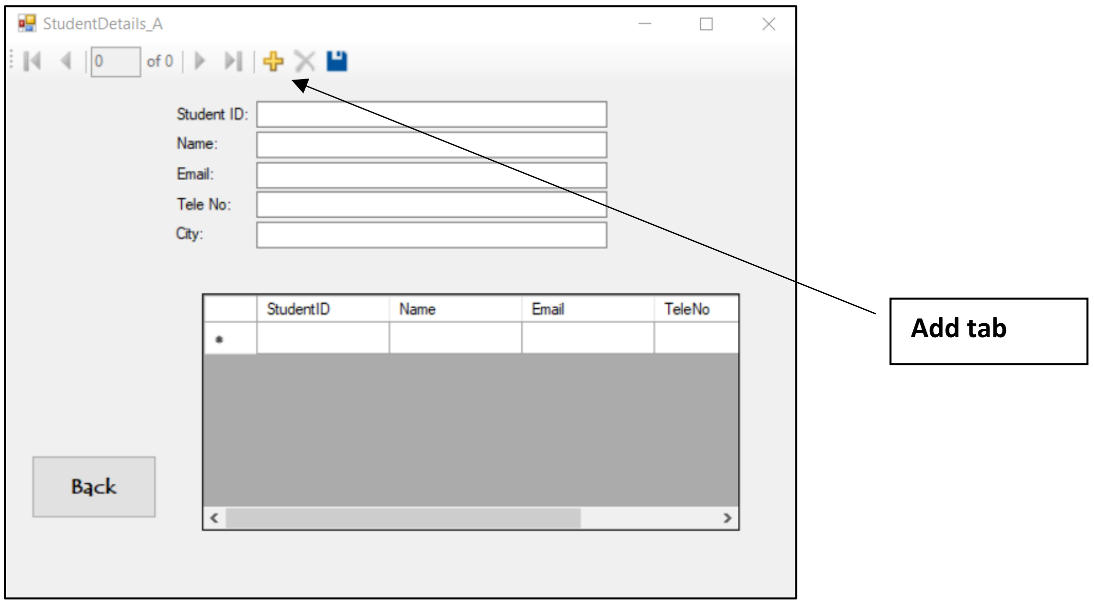
Admin Page **Student Page**



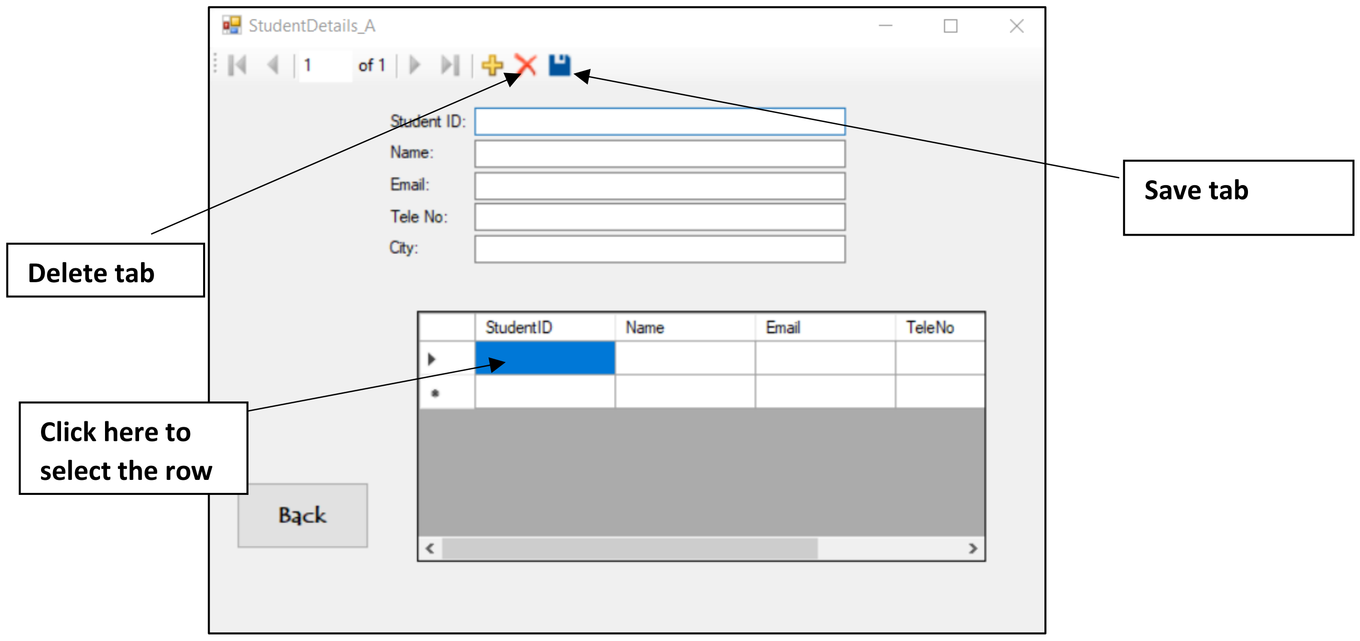


Student Details Button (Admin page)

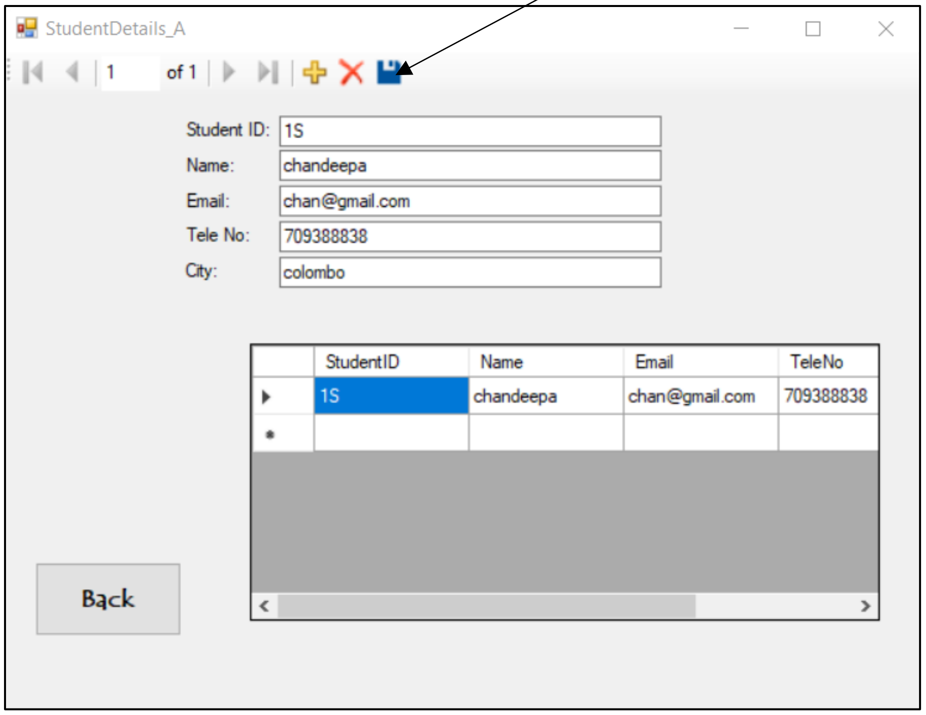
When you click student details button you will get a form shown as below.



In order to enter student details, click "**Add Tab**".   Then the form shown as below.



Then enter Student details in each text boxes given and click **"Save tab"** to save Details in the database.

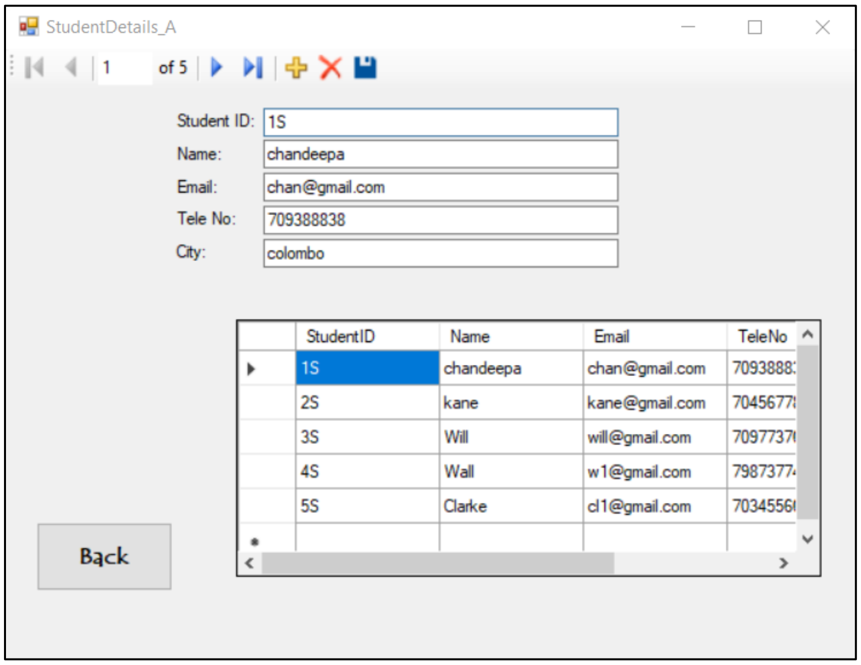


Student ID:	1S
Name:	chandeepa
Email:	chan@gmail.com
Tele No:	709388838
City:	colombo

StudentID	Name	Email	TeleNo
1S	chandeepa	chan@gmail.com	709388838

After entering one student's details, to enter another student's details click **"Add tab"** and enter details in the same way. And click **"Save tab"** to save that data in database.

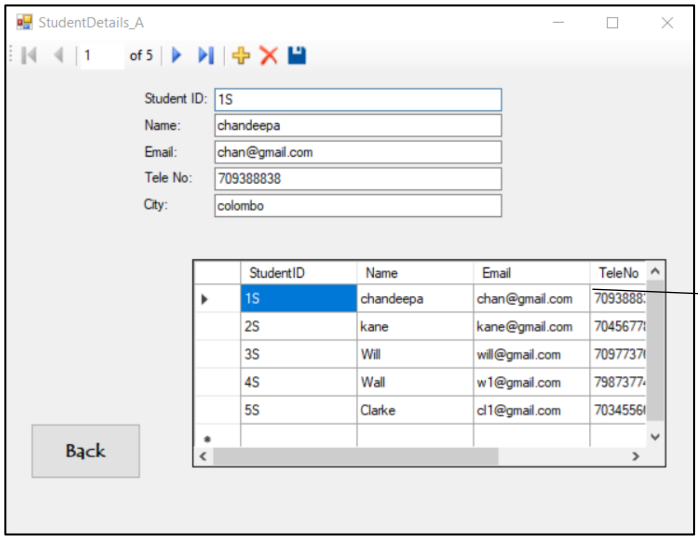
This is how the student details form look like after entering set of data.



Student ID:	1S
Name:	chandeepa
Email:	chan@gmail.com
Tele No:	709388838
City:	colombo

StudentID	Name	Email	TeleNo
1S	chandeepa	chan@gmail.com	7093888
2S	kane	kane@gmail.com	7045677
3S	Will	will@gmail.com	7097737
4S	Wall	w1@gmail.com	7987377
5S	Clarke	cl1@gmail.com	7034556

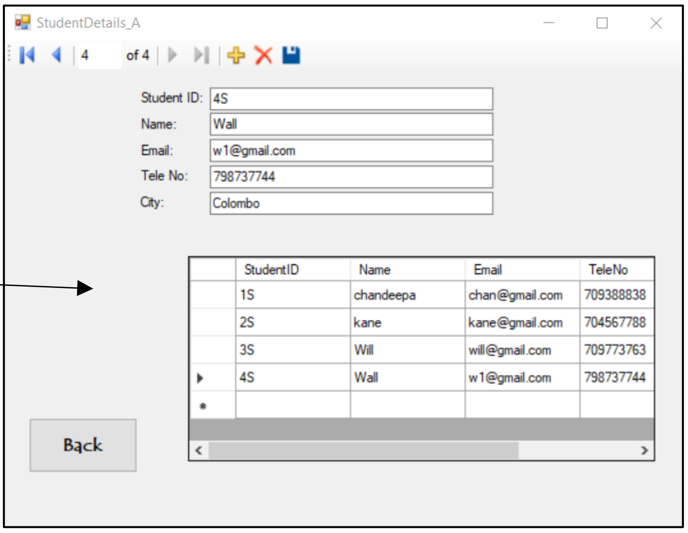
If the user wants to delete entered data, select that specific row and click **"Delete tab"**. After deleting that row data click **"Save tab"** to update the database.



Student ID:	1S
Name:	chandeepa
Email:	chan@gmail.com
Tele No:	709388838
City:	colombo

StudentID	Name	Email	TeleNo
1S	chandeepa	chan@gmail.com	7093888
2S	kane	kane@gmail.com	7045677
3S	Will	will@gmail.com	7097737
4S	Wall	w1@gmail.com	7987377
5S	Clarke	cl1@gmail.com	7034556

Before deleting 5th row data



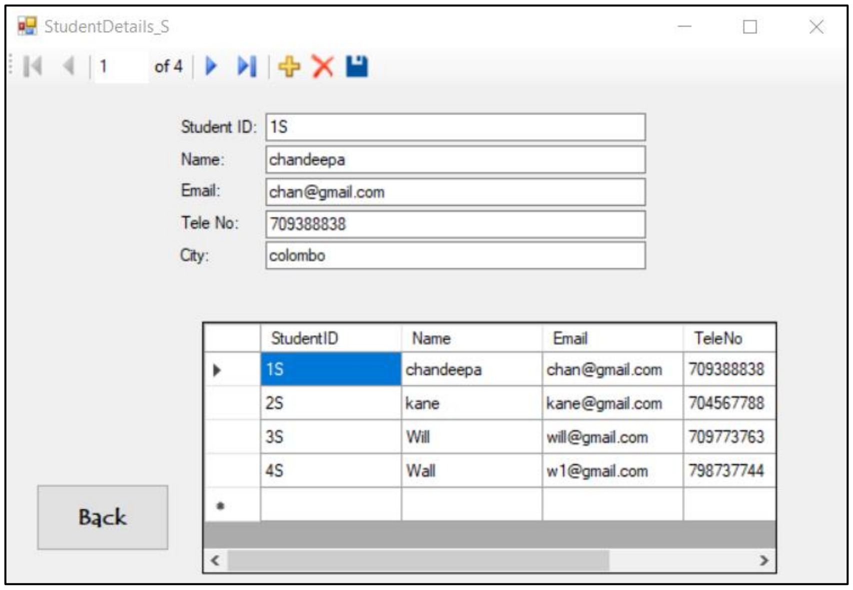
Student ID:	4S
Name:	Wall
Email:	w1@gmail.com
Tele No:	798737744
City:	Colombo

StudentID	Name	Email	TeleNo
1S	chandeepa	chan@gmail.com	709388838
2S	kane	kane@gmail.com	704567788
3S	Will	will@gmail.com	709773763
4S	Wall	w1@gmail.com	798737744

After deleting 5th row data

Student Details Button (Student page)

When you click Student Details button in student page, user will get a same interface form that got from clicking Student Details button in Admin page. This button is also working the same way that the Student Details button in Admin page works.



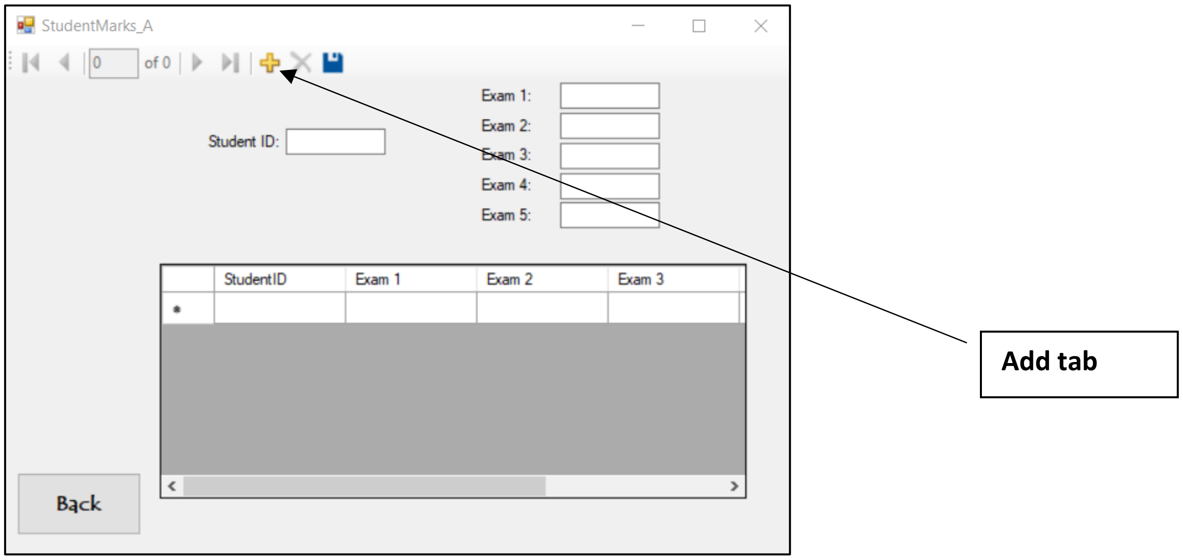
Student ID:	1S
Name:	chandeepa
Email:	chan@gmail.com
Tele No:	709388838
City:	colombo

StudentID	Name	Email	TeleNo
1S	chandeepa	chan@gmail.com	709388838
2S	kane	kane@gmail.com	704567788
3S	Will	will@gmail.com	709773763
4S	Wall	w1@gmail.com	798737744

When admin inputs student data in student details form Students also can view this data. Students also can input, edit& delete data from through this form.

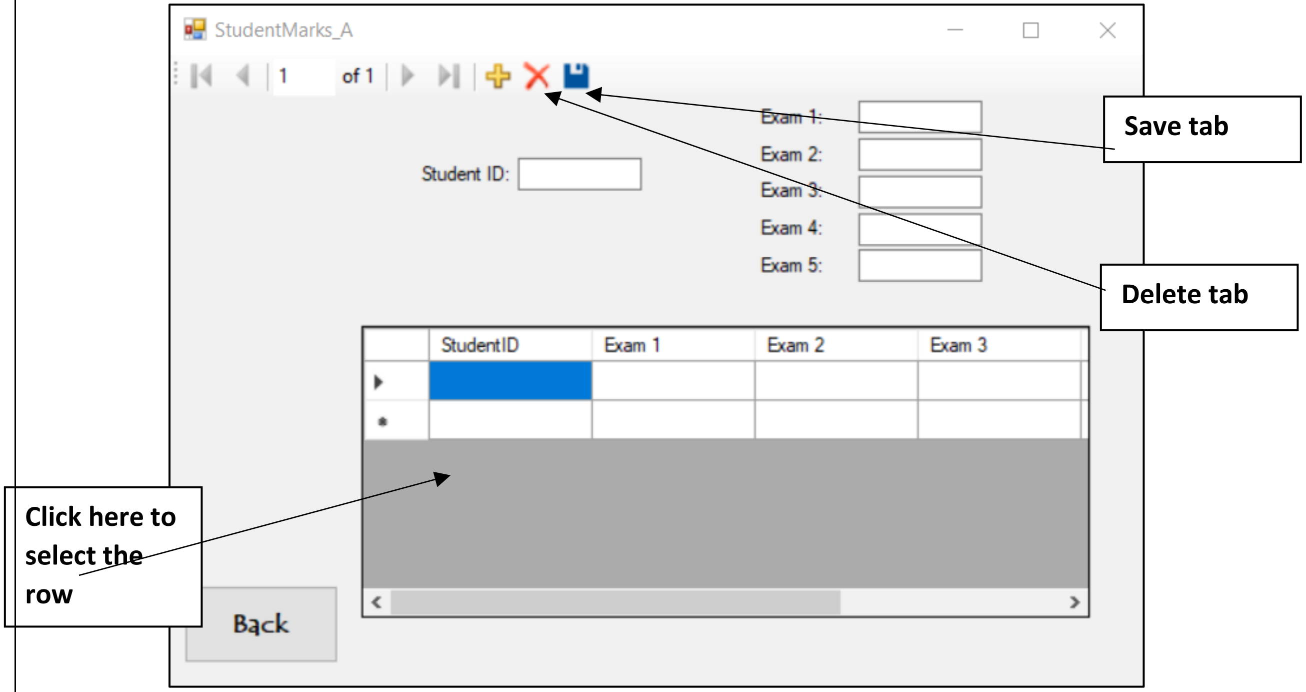
Student Marks Button (Admin page)

When a user click student marks button in admin page user can see a form shown as below.

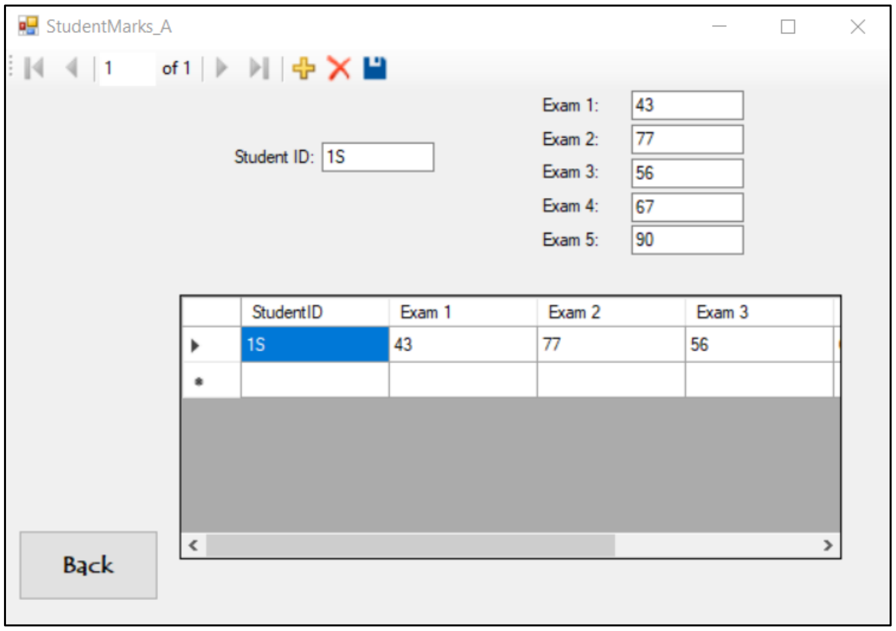


In order to enter student marks, click “**Add Tab**”.

Then the form shown as below.



Then enter the correct Student ID and Student Marks in each text boxes given and click **“Save tab”** to save Details in the database



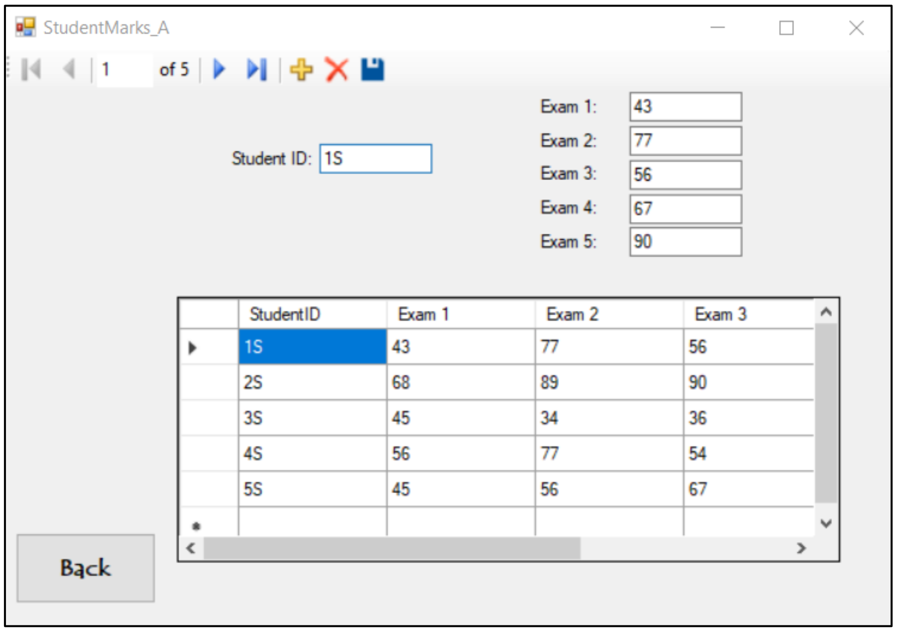
Student ID: 1S

Exam 1: 43
Exam 2: 77
Exam 3: 56
Exam 4: 67
Exam 5: 90

StudentID	Exam 1	Exam 2	Exam 3
1S	43	77	56

After entering one student's marks, to enter another student's marks click **"Add tab"** and enter details in the same way. And click **"Save tab"** to save that data in database.

This is how the student details form look like after entering set of data

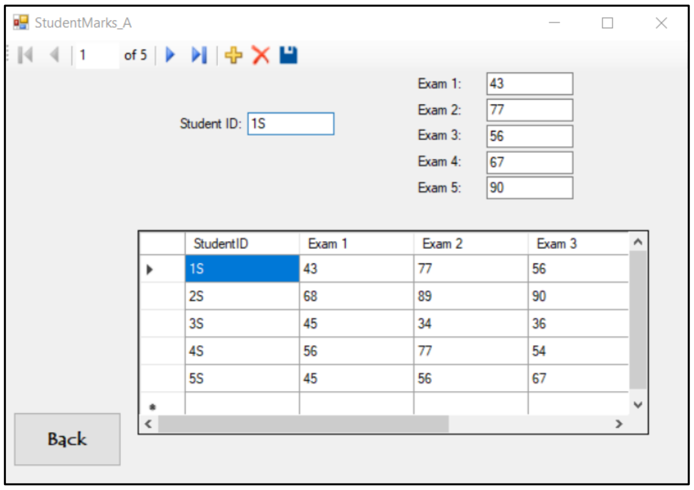


Student ID: 1S

Exam 1: 43
Exam 2: 77
Exam 3: 56
Exam 4: 67
Exam 5: 90

	StudentID	Exam 1	Exam 2	Exam 3
▶	1S	43	77	56
	2S	68	89	90
	3S	45	34	36
	4S	56	77	54
	5S	45	56	67
*				

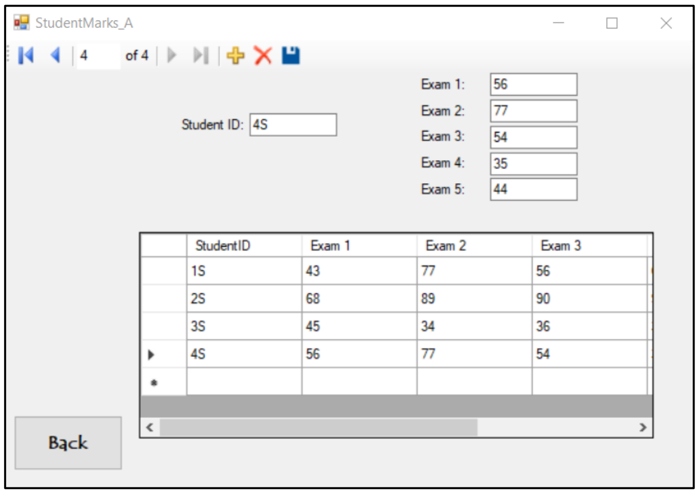
If the user wants to delete entered data, select that specific row and click **"Delete tab"**. After deleting that row data click **"Save tab"** to update the database.



Student ID: 1S

Exam 1: 43
Exam 2: 77
Exam 3: 56
Exam 4: 67
Exam 5: 90

	StudentID	Exam 1	Exam 2	Exam 3
▶	1S	43	77	56
	2S	68	89	90
	3S	45	34	36
	4S	56	77	54
	5S	45	56	67
*				



Student ID: 4S

Exam 1: 56
Exam 2: 77
Exam 3: 54
Exam 4: 35
Exam 5: 44

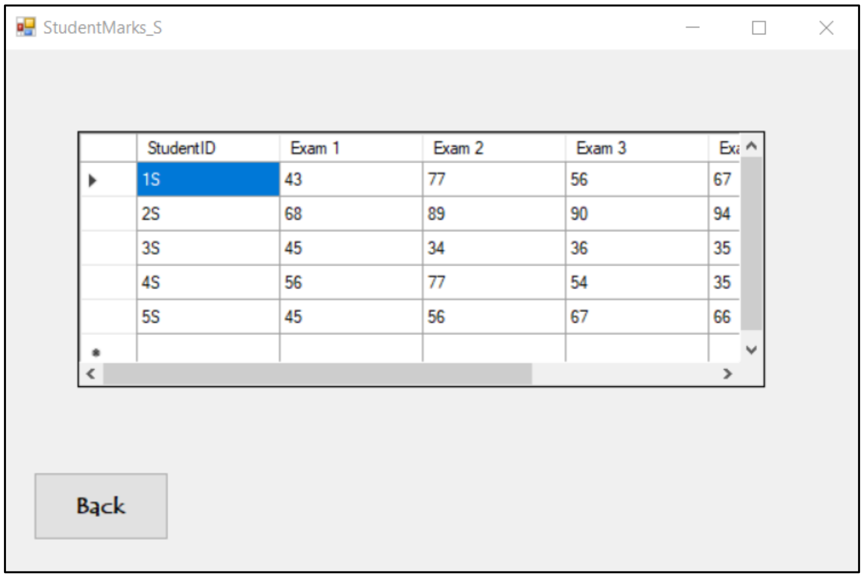
	StudentID	Exam 1	Exam 2	Exam 3
	1S	43	77	56
	2S	68	89	90
	3S	45	34	36
▶	4S	56	77	54
*				

Before deleting 5th row data

After deleting 5th row data

Student Marks Button (Student page)

When a user click student marks button in student page user can see a form shown as below.

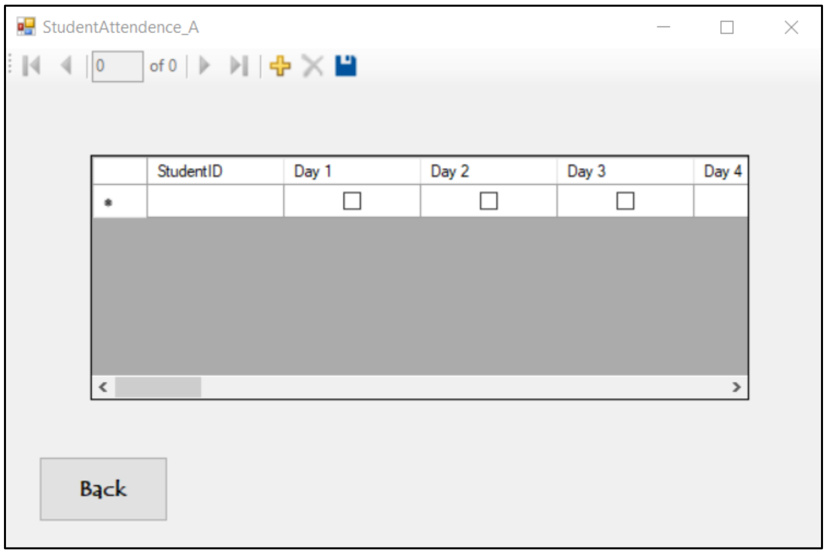


	StudentID	Exam 1	Exam 2	Exam 3	Exa
▸	1S	43	77	56	67
	2S	68	89	90	94
	3S	45	34	36	35
	4S	56	77	54	35
	5S	45	56	67	66
*					

Here students cannot change marks. That can be only done by administration through admin page. So, students can only read their marks.

Student Attendance Button (Admin page)

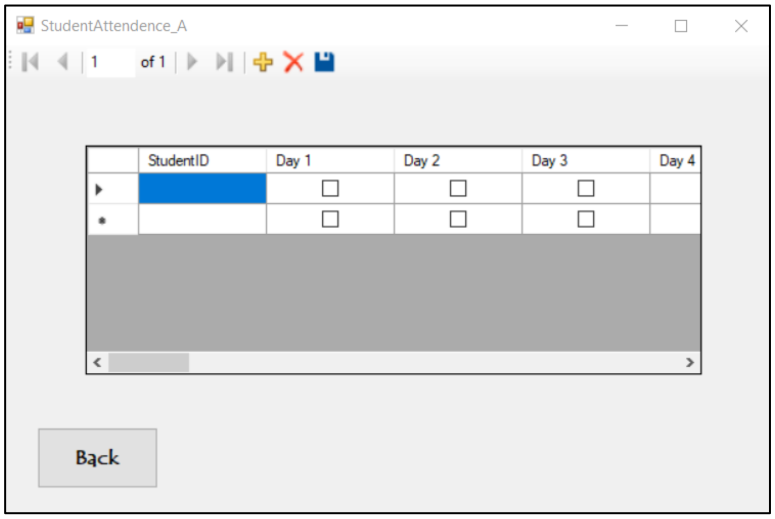
When a user click student attendance button in admin page user can see a form shown as below.



	StudentID	Day 1	Day 2	Day 3	Day 4
*		☐	☐	☐	

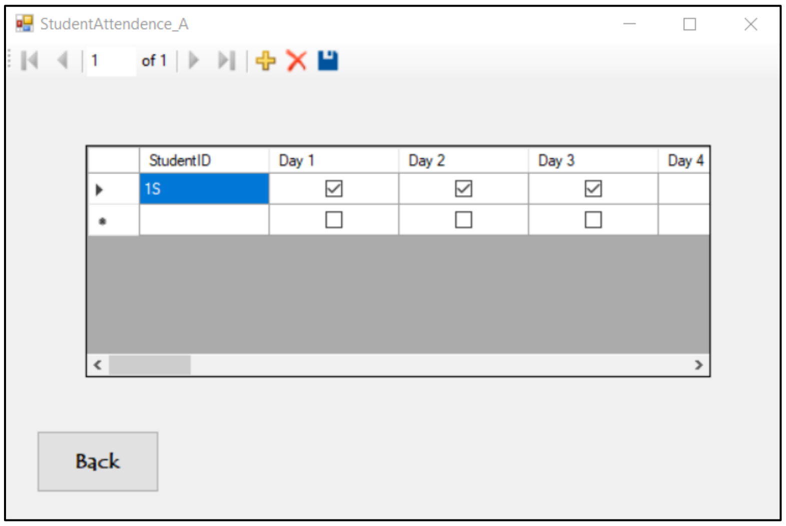
In order to enter student marks, click "**Add Tab**".

Then the form shown as below.



	StudentID	Day 1	Day 2	Day 3	Day 4
▸		☐	☐	☐	
*		☐	☐	☐	

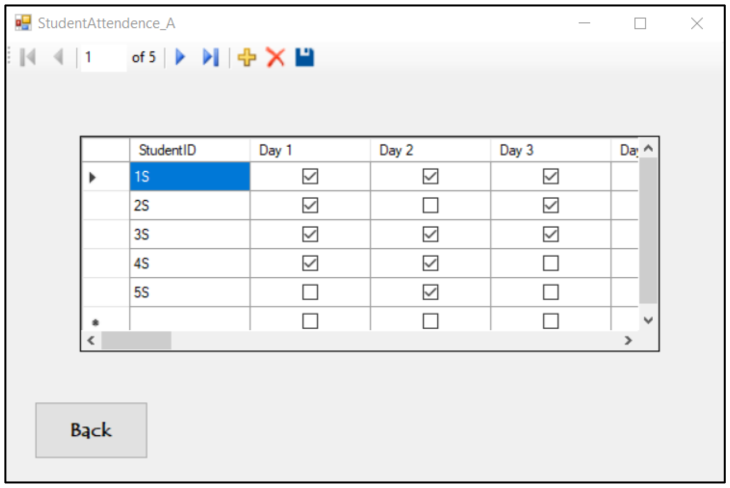
Then mark Student attendance in each check boxes given and click **"Save tab"** to save Details in the database.



	StudentID	Day 1	Day 2	Day 3	Day 4
▸	1S	☑	☑	☑	
*		☐	☐	☐	

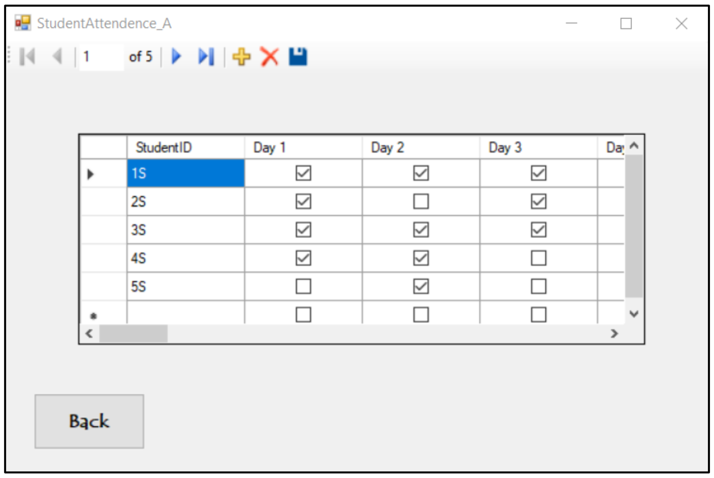
After entering one student's attendance, to enter another student's attendance click **"Add tab"** and enter details in the same way. And click **"Save tab"** to save that data in database.

This is how the student details form look like after entering set of data.



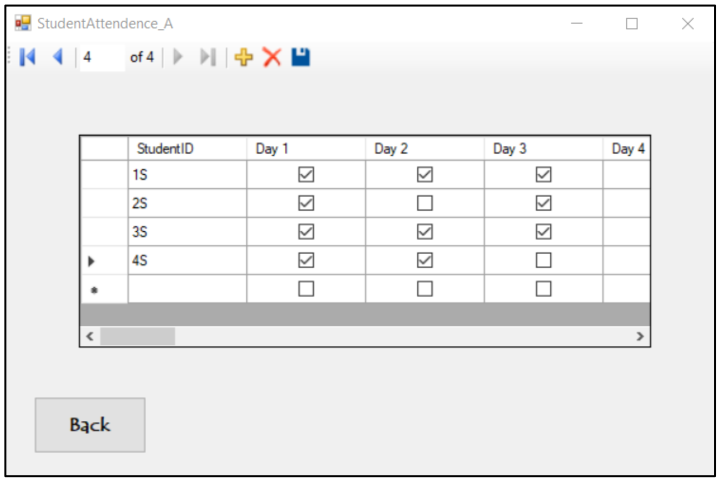
	StudentID	Day 1	Day 2	Day 3	Da
▸	1S	☑	☑	☑	
	2S	☑	☐	☑	
	3S	☑	☑	☑	
	4S	☑	☑	☐	
	5S	☐	☑	☐	
*		☐	☐	☐	

If the user wants to delete entered data, select that specific row and click **“Delete tab”**. After deleting that row data click **“Save tab”** to update the database.



	StudentID	Day 1	Day 2	Day 3	Da
▸	1S	☑	☑	☑	
	2S	☑	☐	☑	
	3S	☑	☑	☑	
	4S	☑	☑	☐	
	5S	☐	☑	☐	
*		☐	☐	☐	

Before deleting 5th row data



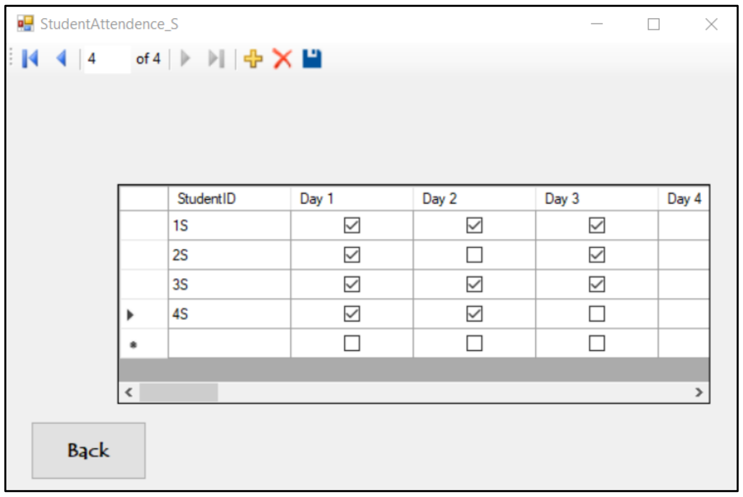
	StudentID	Day 1	Day 2	Day 3	Day 4
	1S	☑	☑	☑	
	2S	☑	☐	☑	
	3S	☑	☑	☑	
▸	4S	☑	☑	☐	
*		☐	☐	☐	

After deleting 5th row data

Student Attendance Button (Student page)

When you click Student Attendance button in student page, user will get the same interface form that got from clicking Student Attendance button in Admin page. This button is also working same way Student Attendance button in Admin page works.

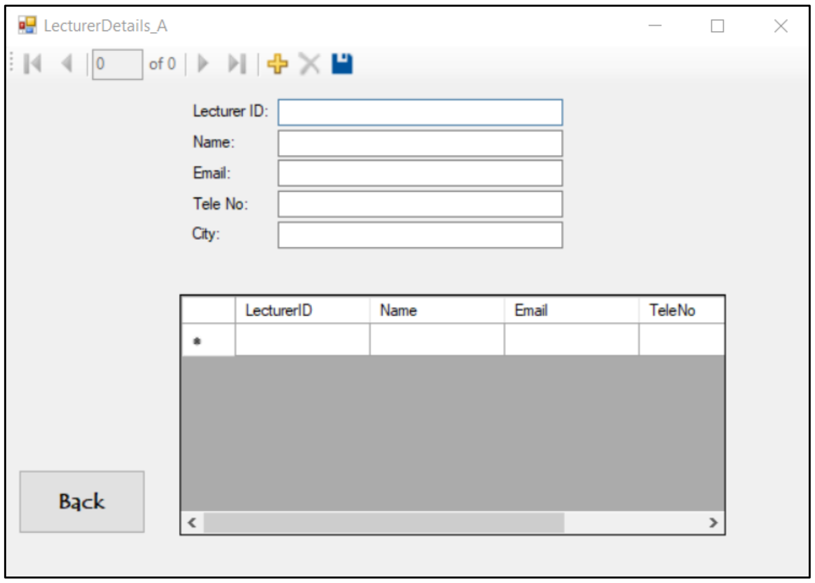
This is how Student Attendance form view after adjustments made by the administration.



	StudentID	Day 1	Day 2	Day 3	Day 4
	1S	☑	☑	☑	
	2S	☑	☐	☑	
	3S	☑	☑	☑	
▸	4S	☑	☑	☐	
*		☐	☐	☐	

Lecturer Details Button (Admin page)

When a user click Lecturer details button in student page user can see a form shown as below.

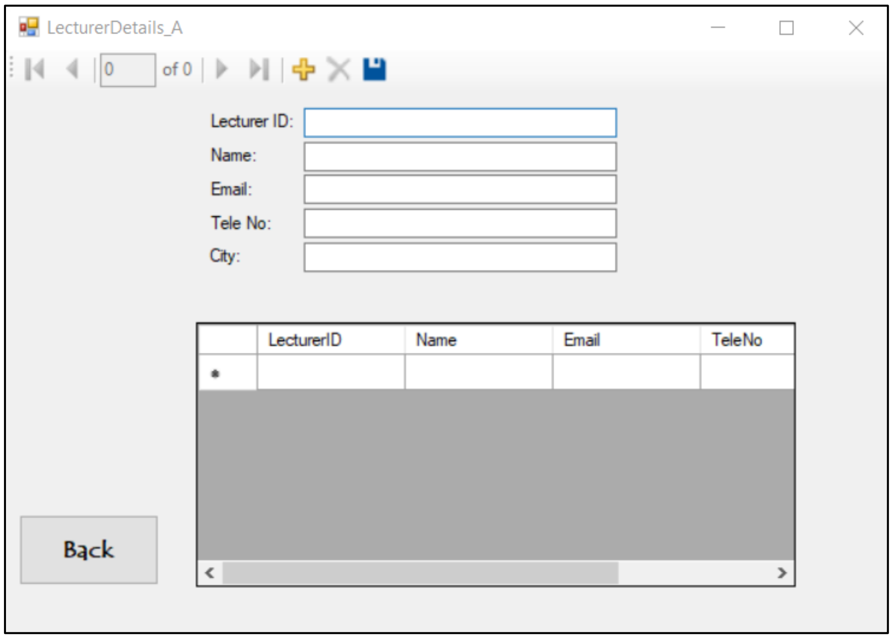


Lecturer ID:	
Name:	
Email:	
Tele No:	
City:	

	LecturerID	Name	Email	TeleNo
*				

In order to enter lecturer details, click "**Add Tab**". Then the form shown as below.

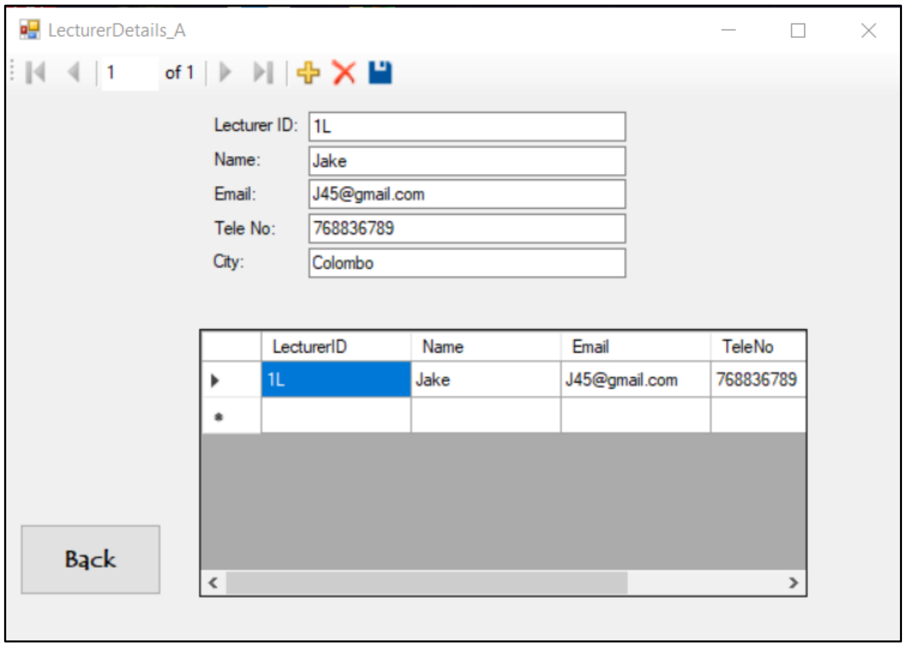
-



Lecturer ID:	
Name:	
Email:	
Tele No:	
City:	

	LecturerID	Name	Email	TeleNo
*				

Then enter Lecturer details in each text boxes given and click **"Save tab"** to save Details in the database.

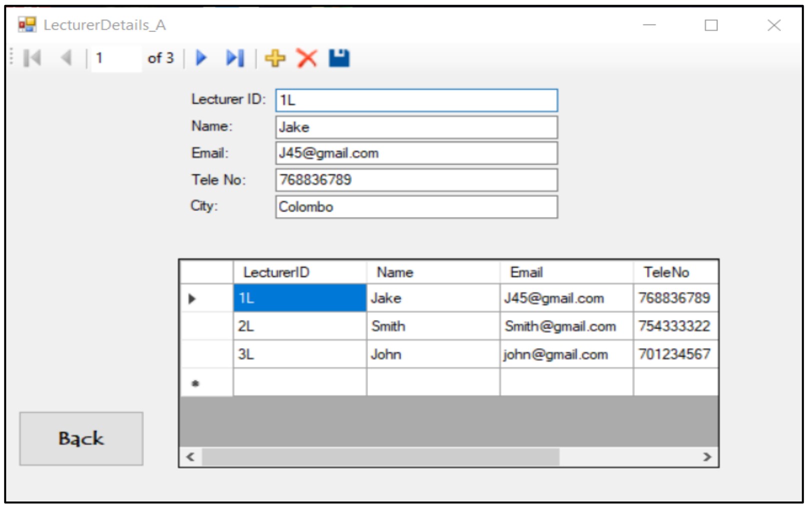


Lecturer ID:	1L
Name:	Jake
Email:	J45@gmail.com
Tele No:	768836789
City:	Colombo

	LecturerID	Name	Email	TeleNo
▶	1L	Jake	J45@gmail.com	768836789
*				

After entering one Lecturer's details, to enter another lecturer's details click **"Add tab"** and enter details in the same way. And click **"Save tab"** to save that data in database.

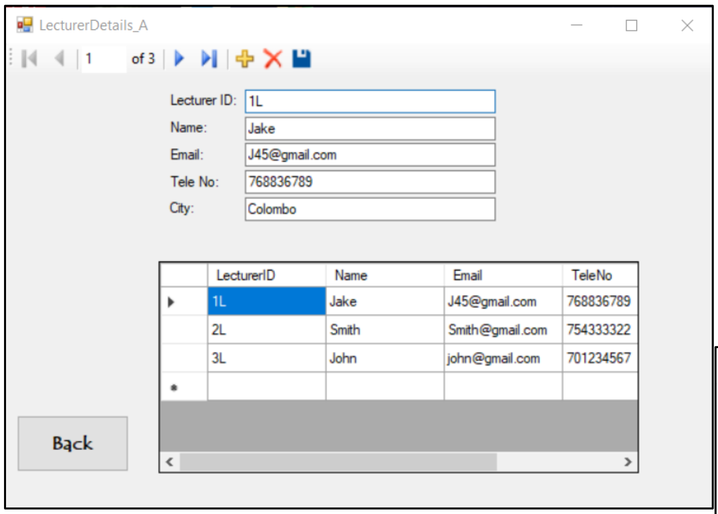
This is how the lecturer details form look like after entering set of data.



Lecturer ID:	1L
Name:	Jake
Email:	J45@gmail.com
Tele No:	768836789
City:	Colombo

LecturerID	Name	Email	TeleNo
1L	Jake	J45@gmail.com	768836789
2L	Smith	Smith@gmail.com	754333322
3L	John	john@gmail.com	701234567

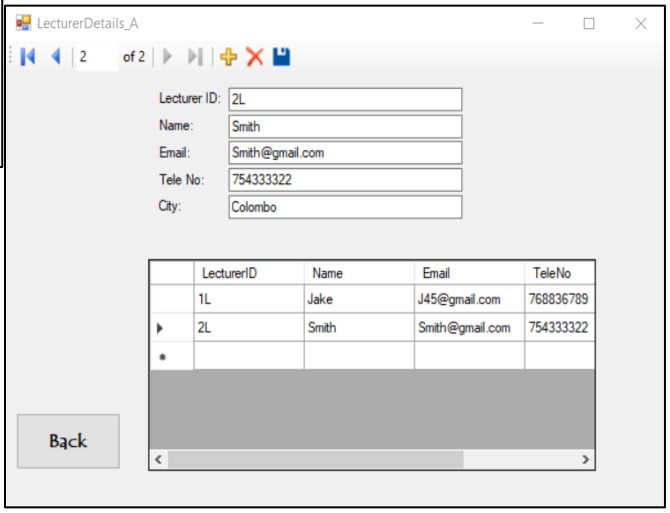
If the user wants to delete entered data, select that specific row and click **"Delete tab"**. After deleting that row data click **"Save tab"** to update the database.



Lecturer ID:	1L
Name:	Jake
Email:	J45@gmail.com
Tele No:	768836789
City:	Colombo

LecturerID	Name	Email	TeleNo
1L	Jake	J45@gmail.com	768836789
2L	Smith	Smith@gmail.com	754333322
3L	John	john@gmail.com	701234567

Before deleting 3rd row data



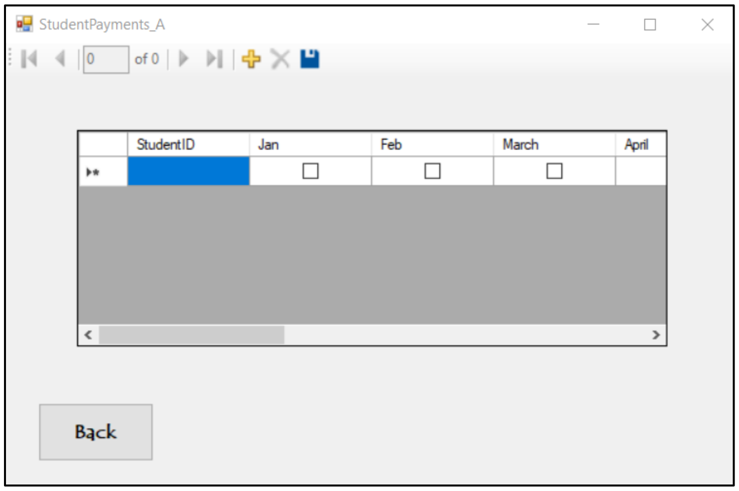
Lecturer ID:	2L
Name:	Smith
Email:	Smith@gmail.com
Tele No:	754333322
City:	Colombo

LecturerID	Name	Email	TeleNo
1L	Jake	J45@gmail.com	768836789
2L	Smith	Smith@gmail.com	754333322

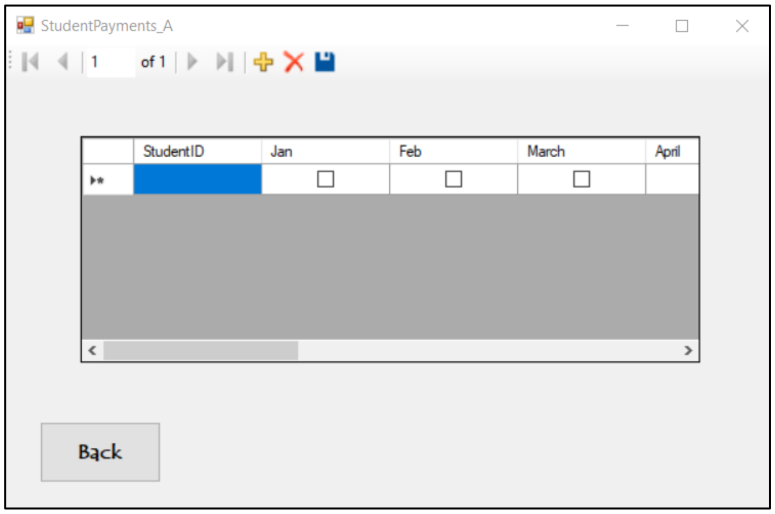
After deleting 3rd row data

Students Payments Button (Admin page)

When a user click student attendance button in admin page user can see a form shown as below.

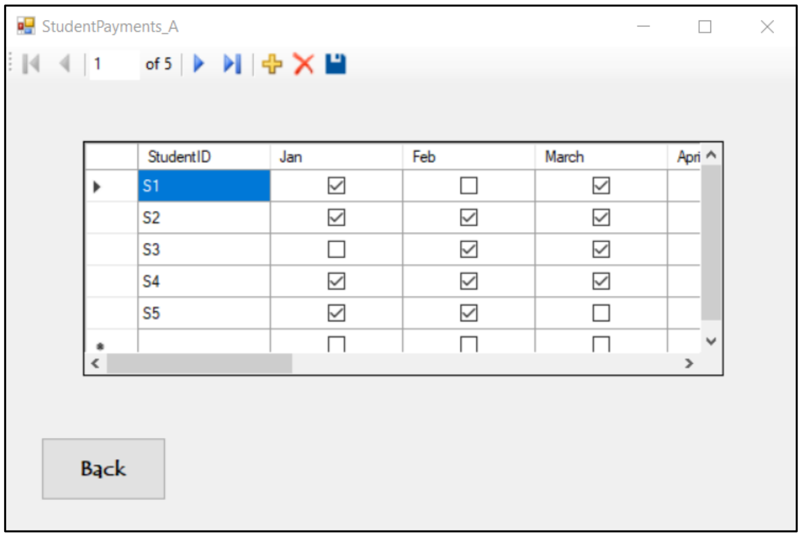


In order to enter student payments, click "**Add Tab**". Then the form shown as below.



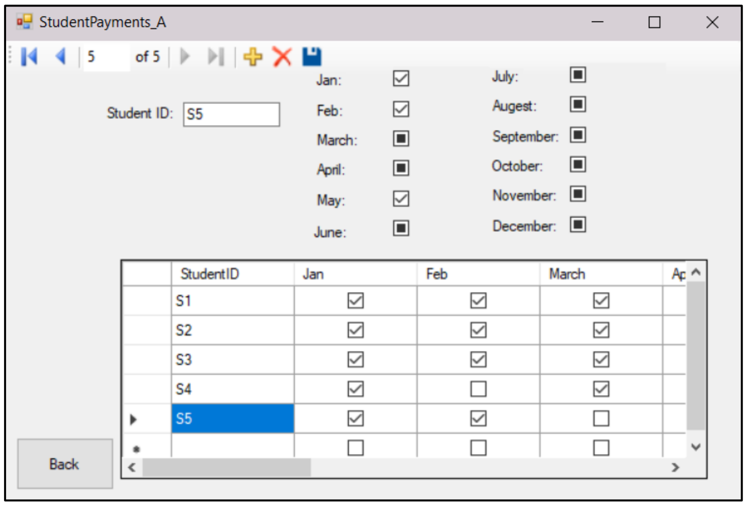
After entering one student's payments, to enter another student's payments click **"Add tab"** and enter details in the same way. And click **"Save tab"** to save that data in database.

This is how the student payments form look like after entering set of data.



	StudentID	Jan	Feb	March	Apri
▸	S1	✓	☐	✓	
	S2	✓	✓	✓	
	S3	☐	✓	✓	
	S4	✓	✓	✓	
	S5	✓	✓	☐	
*		☐	☐	☐	

If the user wants to delete entered data, select that specific row and click **"Delete tab"**. After deleting that row data click **"Save tab"** to update the database.

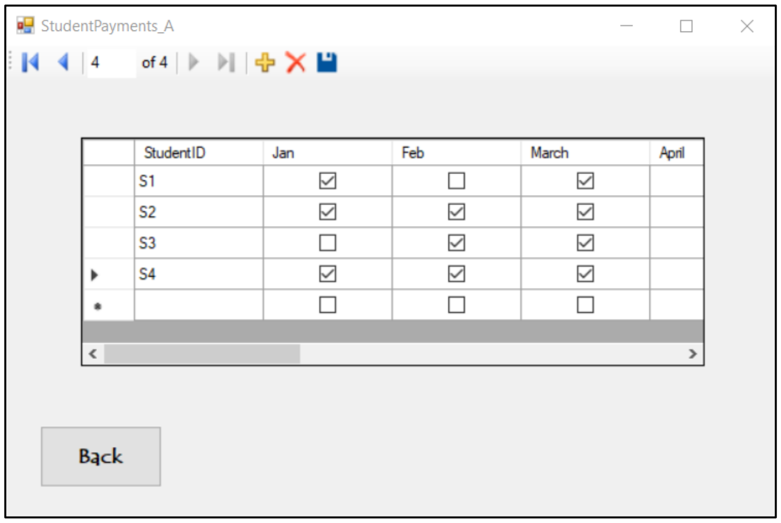


Student ID: S5

Jan:	✓	July:	■
Feb:	✓	Augest:	■
March:	■	September:	■
April:	■	October:	■
May:	✓	November:	■
June:	■	December:	■

	StudentID	Jan	Feb	March	Ap
	S1	✓	✓	✓	
	S2	✓	✓	✓	
	S3	✓	✓	✓	
	S4	✓	☐	✓	
▸	S5	✓	✓	☐	
*		☐	☐	☐	

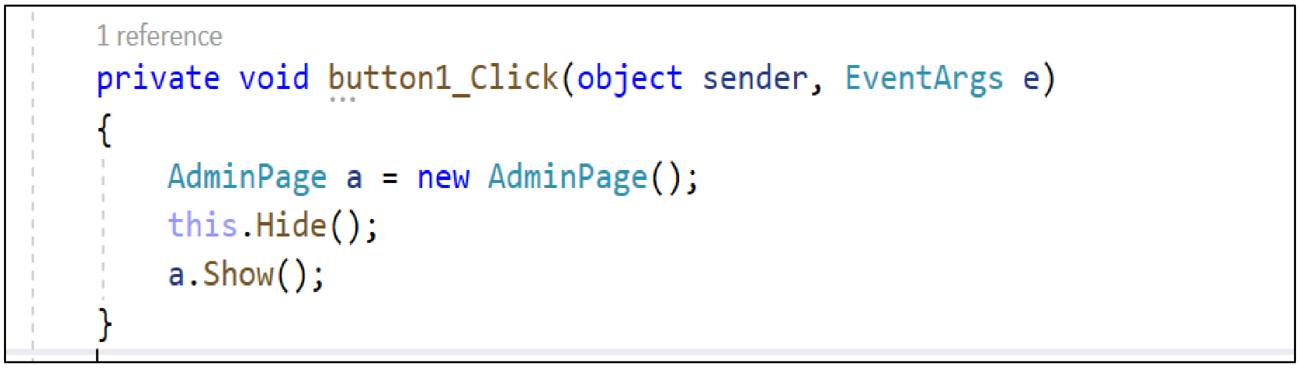
Before deleting 5th row data



	StudentID	Jan	Feb	March	April
	S1	✓	☐	✓	
	S2	✓	✓	✓	
	S3	☐	✓	✓	
▸	S4	✓	✓	✓	
*		☐	☐	☐	

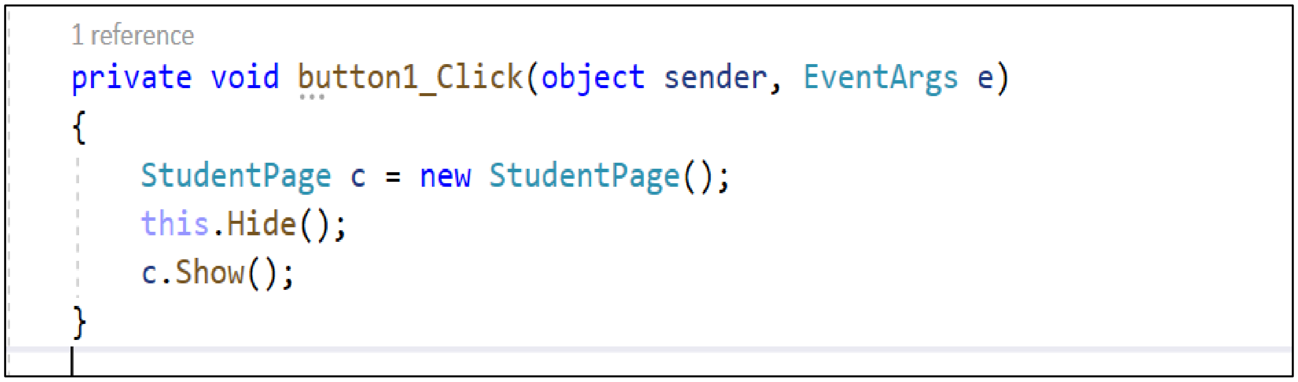
After deleting 5th row data

Back button coding in admin page buttons



```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    AdminPage a = new AdminPage();
    this.Hide();
    a.Show();
}
```

Back button coding in student page buttons



```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    StudentPage c = new StudentPage();
    this.Hide();
    c.Show();
}
```

6.Test Plan

a. Testing Procedure

According to time frame in the project proposal testing procedure has given 30 days.

Time	**Frist Week**	**Second Week**	**Third Week**	**Fourth Week**
Project Tasks				
Project Planning	3 days			
Project Design	3 days	3 days		
Project Coding		3 days	4 days	
Project Testing			3 days	
Project Reporting				2 days

FUNCTIONS	**TIME**
PLANNING	3 days
DESIGNING	6 days
CODING	7 days
TESTING	3 days
REPORTING	2 days

a. Tested Items

All the functions (Login, Insert, Update, Delete, Exit) have been tested and make sure that they are working properly and did not have any kind of errors

6.Software and Hardware Requirements

Software

1.Visual Studio 2017

- Here I used this software to build this Student Management System.
- We used c# as the programming language.
- We used a local database created inside visual studio.

2.Operating System

As the operating system we used Microsoft Windows 10.

Hardware

- Hard drive-512GB
- Processor-Intel core i7(10th gen)
- Ram-4GB
- Laptop-Dell
- Printer-Canon 1505

7.Future Improvements

Our Student Management System Provides important information for administration, students and lectures at ABC institute. This system is a very fast, effective and efficient than the previous manual student management system that they had.

We are also hoping to add new features to our system in the future too.

- Adding a student login system to each and every student
- Lecturer and Student performance evaluation method
- Student feedback receiving method

These are the some of future improvements we are hoping to do for this system.

7.Conclusion

Our project is only a humble venture to satisfy the needs in an institution. Several user-friendly coding also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the organization. This new system will help students , administration and lecturers to improve their efficiency and effectiveness.