

AYYA NADAR JANAKI AMMAL COLLEGE (Autonomous),Sivakasi

DEPARTMENT OF COMPUTER SCIENCE (UG)

(2021-2022)



RECORD WORK

S5C16-PRACTICAL IV (.NET & ANDROID LAB)

NAME : S. ARUNPRIYA

REG.NO : 19US03

This is to certify to be the bonafide work of the student in this department.

Place: Sivakasi

Faculty in-charge

Date: 22.11.2021

(M.SAKTHI SARAVANAN)

INTERNAL MARK

Submitted for the Practical Examination held on 03.12.2021.

INTERNAL EXAMINER

EXTERNAL EXAMINER

Ex.No	Date	Programs	Page No	Sign
Console Application				
1.		Exception Handling	05	
2.		Interface	08	
3.		Inheritance	12	
4.		Polymorphism	17	
5.		Operator Overloading	22	
6.		Sorting the numbers in ascending order	25	
7.		Display the minimum and maximum numbers array	28	
8.		Display the Tables from 1 to 10	30	
9.		Display the calculator	32	
Windows Form Application				
10.		Agriculture Management System	66	
11.		Cargo Management System	79	
12.		Bike Insurance Management System	89	
13.		Voting Management System	97	
14.		Software Learning System	105	
Web Application				
15.		Login Page in ASP.NET	117	
16.		Validation Controls	125	
17.		Mark Statement	133	
			117	

CONSOLE APPLICATION

A console is an operating system window through which a user can communicate with the operating system or we can say a console is an application in which we can give text as an input from the keyboard and get the text as an output from the computer end. The command prompt is an example of a console in the windows and which accept MS-DOS commands. The console contains two attributes named as screen buffer and a console window.

In C#, the `Console` class is used to represent the standard input, output, and error streams for the console applications. You are not allowed to inherit `Console` class. This class is defined under `System` namespace. This class does not contain any constructor. Instead of the constructor, this class provides different types of properties and methods to perform operations.

- **Console.WriteLine()** – writes the specified word to the console window.
- **Console.ReadLine()** – This does the same thing but adds a new line character at the end of the output.

EXAMPLE :

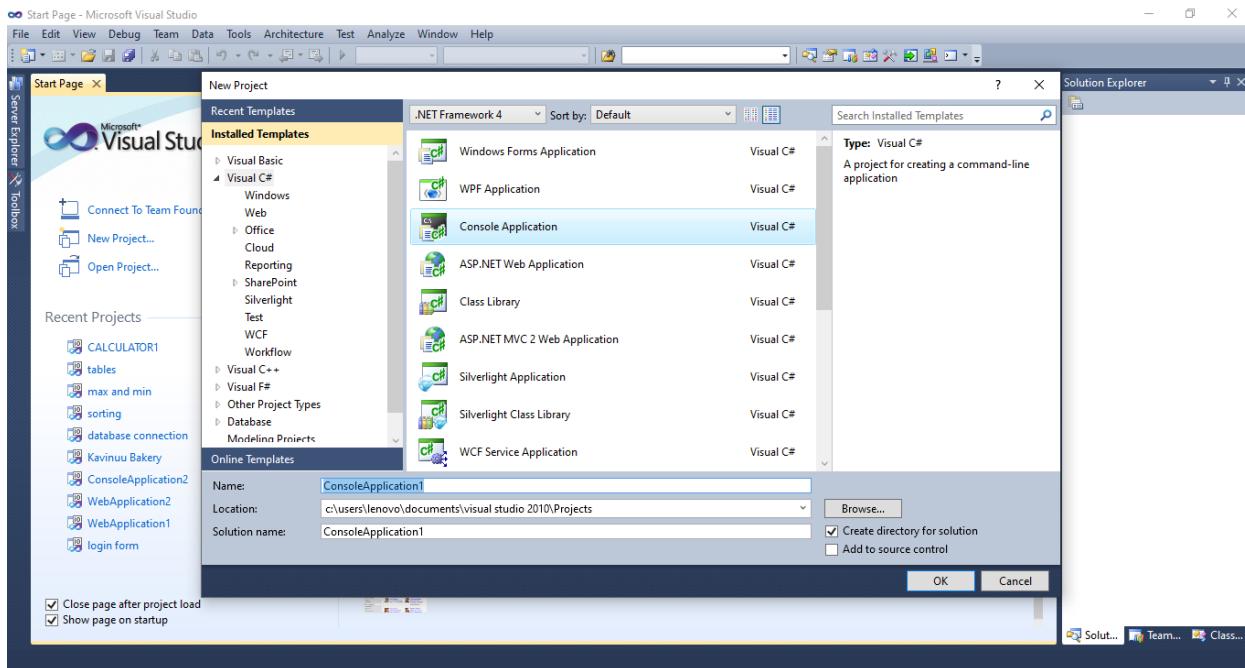
The below C# coding lets the user input a line of text and displays that text.

```
String s=Console.ReadLine();
```

```
Console.WriteLine(s);
```

To write C# console program,

Open visual studio->file->New Project->Visual C#->console applications.



EXCEPTION HANDLING

An exception is a problem that arises during the execution of a program. A C# exception is a response to an exceptional circumstance that arises while a program is running, such as an attempt to divide by zero.

Exceptions provide a way to transfer control from one part of a program to another. C# exception handling is built upon four keywords: **try**, **catch**, **finally**, and **throw**.

- **try** – A try block identifies a block of code for which particular exceptions are activated. It is followed by one or more catch blocks.
- **catch** – A program catches an exception with an exception handler at the place in a program where you want to handle the problem. The catch keyword indicates the catching of an exception.

- **finally** – The finally block is used to execute a given set of statements, whether an exception is thrown or not thrown. For example, if you open a file, it must be closed whether an exception is raised or not.
- **throw** – A program throws an exception when a problem shows up. This is done using a throw keyword.

Syntax

Assuming a block raises an exception, a method catches an exception using a combination of the try and catch keywords. A try/catch block is placed around the code that might generate an exception. Code within a try/catch block is referred to as protected code, and the syntax for using try/catch looks like the following

```
try {
    // statements causing exception
} catch(ExceptionName e1 ) {
    // error handling code
} catch(ExceptionName e2 ) {
    // error handling code
} catch(ExceptionName ) {
    // error handling code
} finally {
    // statements to be executed
}
```



You can list down multiple catch statements to catch different type of exceptions in case your try block raises more than one exception in different situations.

Exception Classes in C#

C# exceptions are represented by classes.

The exception classes in C# are mainly directly or indirectly derived from the **System.Exception** class. Some of the exception classes derived

from the **System.Exception** class are the **System.ApplicationException** and **System.SystemException** classes.

. The **System.ApplicationException** class supports exceptions generated by application programs. Hence the exceptions defined by the programmers should derive from this class.

The **System.SystemException** class is the base class for all predefined system exception.

Sr.No.	Exception Class & Description
1.	System.IO.IOException Handles I/O errors.
2.	System.IndexOutOfRangeException Handles errors generated when a method refers to an array index out of range.
3.	System.ArrayTypeMismatchException Handles errors generated when type is mismatched with the array type.
4.	System.NullReferenceException Handles errors generated from referencing a null object.
5.	System.DivideByZeroException Handles errors generated from dividing a dividend with zero.
6.	System.InvalidCastException Handles errors generated during typecasting.
7.	System.OutOfMemoryException Handles errors generated from insufficient free memory.

EX.NO:1

Date:

EXCEPTION HANDLING

AIM :

To create a console application using exception handling in C#

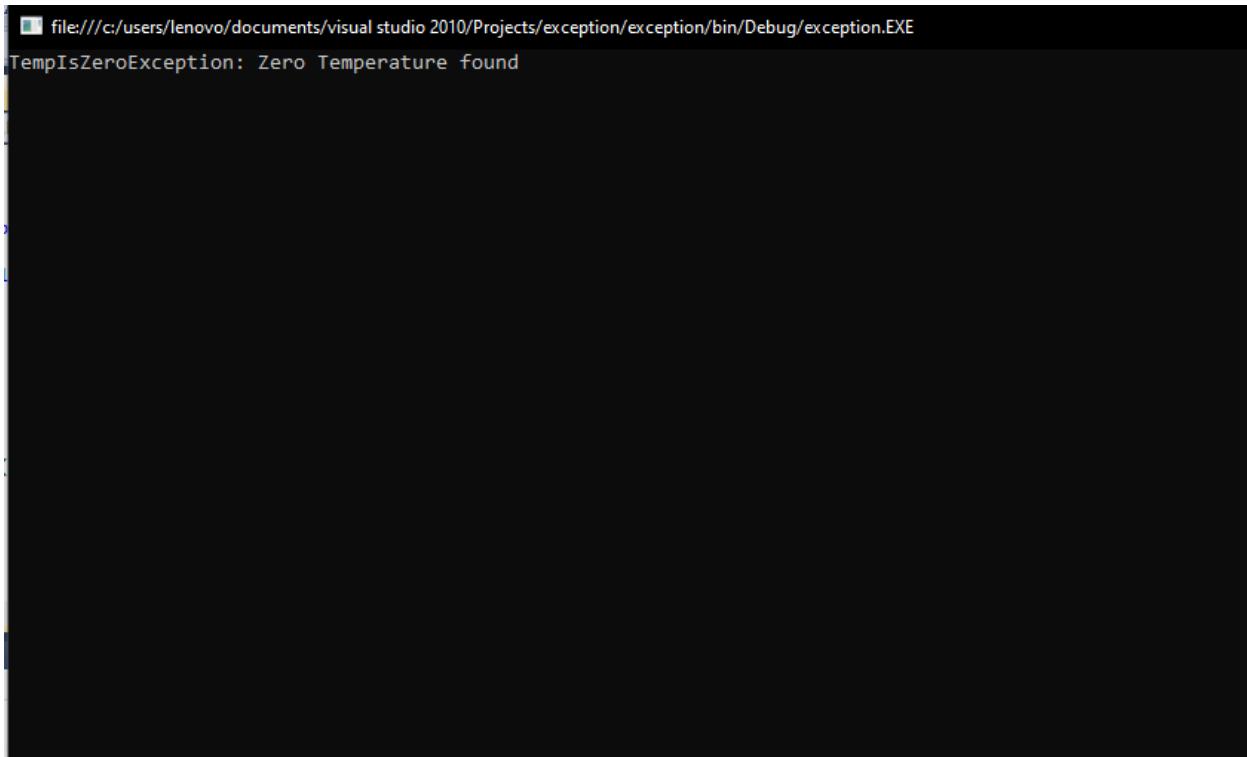
PROGRAM :

```
usingSystem;
namespaceUserDefinedException{
classTestTemperature{
static void Main(string[]args){
Temperature temp =newTemperature();
try{
temp.showTemp();
}
catch(TempIsZeroException e){
Console.WriteLine("TempIsZeroException: {0}",e.Message);
}
Console.ReadKey();
}
}
publicclassTempIsZeroException:Exception{
publicTempIsZeroException(string message):base(message){
}
}
publicclassTemperature{
int temperature =0;
publicvoidshowTemp(){
if(temperature ==0){
throw(newTempIsZeroException("Zero Temperature found"));
}
}
}
```



```
 }else{
    Console.WriteLine("Temperature: {0}", temperature);
}
}
```

OUTPUT:



INTERFACES

An interface is defined as a syntactical contract that all the classes inheriting the interface should follow. The interface defines the '**what**' part of the syntactical contract and the deriving classes define the '**how**' part of the syntactical contract.

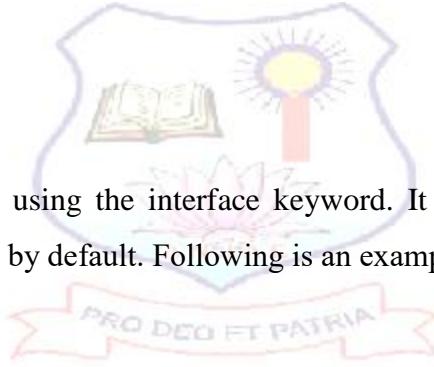
Interfaces define properties, methods, and events, which are the members of the interface. Interfaces contain only the declaration of the members. It is the responsibility of the deriving class to define the members. It often helps in providing a standard structure that the deriving classes would follow.

Abstract classes to some extent serve the same purpose, however, they are mostly used when only few methods are to be declared by the base class and the deriving class implements the functionalities.

Declaring Interfaces

Interfaces are declared using the interface keyword. It is similar to class declaration. Interface statements are public by default. Following is an example of an interface declaration –

```
public interface ITransactions{  
    // interface members  
    void showTransaction();  
    double getAmount();  
}
```



EX.NO:2

Date:

INTERFACES

AIM :

To create a console application using interface in C#.

PROGRAM :

```
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Text;
usingSystem;
namespaceInterfaceApplication{
publicinterfaceITransactions{
voidshowTransaction();
doublegetAmount();
}

publicclassTransaction:ITransactions{
privatestringtCode;
privatestring date;
privatedouble amount;
publicTransaction(){
tCode=" ";
date=" ";
amount=0.0;
}
publicTransaction(string c,string d,double a){
tCode= c;
date= d;
amount= a;
}
publicdoublegetAmount(){
```



```
return amount;  
}  
  
public void showTransaction(){  
    Console.WriteLine("Transaction: {0}", tCode);  
    Console.WriteLine("Date: {0}", date);  
    Console.WriteLine("Amount: {0}", getAmount());  
}  
}  
  
class Tester{  
  
    static void Main(string[] args){  
        Transaction t1 = new Transaction("001", "8/10/2012", 78900.00);  
        Transaction t2 = new Transaction("002", "9/10/2012", 451900.00);  
        t1.showTransaction();  
        t2.showTransaction();  
        Console.ReadKey();  
    }  
}
```



OUTPUT:

```
file:///c:/users/kalimuthu/documents/visual studio 2010/Projects/ConsoleApplication6/ConsoleApplication6/bin/Debu  
TRANSACTION DETAILS  
Transaction: 001  
Date: 8/10/2012  
Amount: 78900  
Transaction: 002  
Date: 9/10/2012  
Amount: 451900
```



INHERITANCE

One of the most important concepts in object-oriented programming is inheritance. Inheritance allows us to define a class in terms of another class, which makes it easier to create and maintain an application. This also provides an opportunity to reuse the code functionality and speeds up implementation time.

When creating a class, instead of writing completely new data members and member functions, the programmer can designate that the new class should inherit the members of an existing class. This existing class is called the **base** class, and the new class is referred to as the **derived** class.

The idea of inheritance implements the **IS-A** relationship. For example, mammal **IS-A** animal, dog **IS-A** mammal hence dog **IS-A** animal as well, and so on.

Base and Derived Classes

A class can be derived from more than one class or interface, which means that it can inherit data and functions from multiple base classes or interfaces.

The syntax used in C# for creating derived classes is as follows –

SYNTAX

```
<acess-specifier> class <base_class> {  
    ...  
}  
  
class<derived_class> : <base_class> {  
    ...  
}
```



EX.NO:3

Date:

INHERITANCE

AIM :

To create a console application using inheritance in C#.

PROGRAM :

```
using System;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Text;
```

namespace inheritance

```
{
    classstd
    {
        publicint m1, m2, m3;
        public string id, name;
        public void getdata()
        {
            try
            {
                Console.WriteLine("enter student id: ");
                id = Console.Read
                Line().ToString();
                Console.WriteLine("enter student name:");
                name = Console.ReadLine().ToString();
                Console.WriteLine("enter the mark in C:");
                m1 = Convert.ToInt32(Console.ReadLine());
                Console.WriteLine("enter the mark in C++:");
                m2 = Convert.ToInt32(Console.ReadLine());
```



```

Console.WriteLine("enter the mark in java: ");
    m3 = Convert.ToInt32(Console.ReadLine());
}
catch (Exception e)
{
    Console.WriteLine(e);
}

class stud : std
{
    public void disp()
    {
        inttotal,avg;
        total=m1+m2+m3;
        avg=total/3;
        Console.WriteLine("STU_ID:{0}",id);
        Console.WriteLine("STU_NAME:{0}",name);
        Console.WriteLine("C MARK:{0}",m1);
        Console.WriteLine("C++ MARK:{0}",m2);
        Console.WriteLine("JAVA MARK:{0}",m3);
        Console.WriteLine("TOTAL:{0}",total);
        Console.WriteLine("AVERAGE:{0}",avg);
        Console.WriteLine("RESULT");
        if((m1>39)&& (m2>39)&& (m3>39))
        {
            if (avg> 90)
                Console.WriteLine("HONOUR");
            else if ((avg> 80) && (avg<= 80))
                Console.WriteLine("DISTINCTION");
        }
    }
}

```



```

else if ((avg> 60) && (avg<= 60))
Console.WriteLine("FIRST CLASS");
else if ((avg> 50) && (avg<= 50))
Console.WriteLine("SECOND CLASS");
else if ((avg> 40) && (avg<= 40))
Console.WriteLine("THIRD CLASS");
}
else
Console.WriteLine("FAIL");
}
}

classstude : stud
{
static void Main()
{
stud s = new stud();
s.getdata();
s.disp();
Console.ReadLine();
}
}

```



OUTPUT :

```
□ Select file:///c:/users/kalimuthu/documents/visual studio 2010/Projects/inheritance/inheritance/bin/Debug/inheritanc
enter student id:
001
enter student name:
PRIYADHARSHINI
enter the mark in C:
89
enter the mark in C++:
90
enter the mark in java:
96
STU_ID:001
STU_NAME:PRIYADHARSHINI
C MARK:89
C++ MARK:90
JAVA MARK:96
TOTAL:275
AVERAGE:91
RESULT
HONOUR
```

POLYMORPHISM

The word **polymorphism** means having many forms. In object-oriented programming paradigm, polymorphism is often expressed as 'one interface, multiple functions'.

Polymorphism can be static or dynamic. In **static polymorphism**, the response to a function is determined at the compile time. In **dynamic polymorphism**, it is decided at run-time.

Static Polymorphism

The mechanism of linking a function with an object during compile time is called early binding. It is also called static binding. C# provides two techniques to implement static polymorphism. They are –

- Function overloading
- Operator overloading



Function Overloading

You can have multiple definitions for the same function name in the same scope. The definition of the function must differ from each other by the types and/or the number of arguments in the argument list. You cannot overload function declarations that differ only by return type.

OPERATOR OVERLOADING

You can redefine or overload most of the built-in operators available in C#. Thus a programmer can use operators with user-defined types as well. Overloaded operators are functions with special names the keyword **operator** followed by the symbol for the operator being defined. similar to any other function, an overloaded operator has a return type and a parameter list.

EX.NO:4

Date:

POLYMORPHISM

AIM :

To create a console application using polymorphism in C#.

PROGRAM:

```
usingSystem;
namespacePolymorphismApplication{
classShape{
protectedint width, height;
publicShape(int a =0,int b =0){
width= a;
height= b;
}
publicvirtualint area(){
Console.WriteLine("Parent class area :");
return0;
}
}
classRectangle:Shape{
publicRectangle(int a =0,int b =0):base(a, b){
}
publicoverrideint area (){
Console.WriteLine("Rectangle class area :");
return(width * height);
}
}
classTriangle:Shape{
publicTriangle(int a =0,int b =0):base(a, b){
}
publicoverrideint area(){
}
```



```

Console.WriteLine("Triangle class area :");
return(width * height /2);
}
}

classCaller{
publicvoidCallArea(Shapesh){
int a;
a =sh.area();
Console.WriteLine("Area: {0}", a);
}
}

classTester{
staticvoidMain(string[]args){
Caller c =newCaller();
Rectangle r =newRectangle(10,7);
Triangle t =newTriangle(10,5);
c.CallArea(r);
c.CallArea(t);
Console.ReadKey();
}
}
}

```



OUTPUT:

```
file:///c:/users/lenovo/documents/visual studio 2010/Projects/polymorphism/polymorphism/bin/Debug/polymorphism.EXE
Rectangle class area :
Area: 70
Triangle class area :
Area: 25
```



OPERATOR OVERLOADING

You can redefine or overload most of the built-in operators available in C#. Thus a programmer can use operators with user-defined types as well. Overloaded operators are functions with special names the keyword **operator** followed by the symbol for the operator being defined. similar to any other function, an overloaded operator has a return type and a parameter list.

EXAMPLE

```
public static Box operator+(Box b, Box c){  
    Box box = new Box();  
    box.length = b.length + c.length;  
    box.breadth = b.breadth + c.breadth;  
    box.height = b.height + c.height;  
    return box;  
}
```



Overloadable and Non-Overloadable Operators

The following table describes the overload ability of the operators in C# –

Sr.No.	Operators & Description
1	$+$, $-$, $!$, \sim , $++$, $--$ These unary operators take one operand and can be overloaded.
2	$+$, $-$, $*$, $/$, $\%$ These binary operators take one operand and can be overloaded.
3	$==$, $!=$, $<$, $>$, $<=$, $>=$ The comparison operators can be overloaded.

4	&&, The conditional logical operators cannot be overloaded directly.
5	+ =, - =, * =, / =, % = The assignment operators cannot be overloaded.
6	=, ., ?:, ->, new, is, sizeof, typeof These operators cannot be overloaded.



EX.NO:5

Date:

OPERATOR OVERLOADING

AIM:

To create a console application using operator overloading in C#.

PROGRAM:

```
usingSystem;
```

```
namespaceOperatorOvlApplication{
```

```
classBox{
```

```
    privatedouble length;
```

```
    privatedouble breadth;
```

```
    privatedouble height;
```

```
    publicdoublegetVolume(){
```

```
        return length * breadth * height;
```

```
}
```

```
    publicvoidsetLength(doublelen){
```

```
        length=len;
```

```
}
```

```
    publicvoidsetBreadth(doublebre){
```

```
        breadth=bre;
```

```
}
```

```
    publicvoidsetHeight(doublehei){
```

```
        height=hei;
```

```
}
```

```
    publicstaticBoxoperator+(Box b,Box c){
```

```
        Boxbox=newBox();
```

```
        box.length=b.length+c.length;
```



```

box.height=b.height+c.height;
box.breadth=b.breadth+c.breadth;
return box;
}
}

class Tester{
static void Main(string[] args){
Box Box1=new Box();
Box Box2=new Box();
Box Box3=new Box();
double volume =0.0;
Box1.setLength(6.0);
Box1.setBreadth(7.0);
Box1.setHeight(5.0);

Box2.setLength(12.0);
Box2.setBreadth(13.0);
Box2.setHeight(10.0);

volume=Box1.getVolume();
Console.WriteLine("Volume of Box1 : {0}", volume);
volume=Box2.getVolume();
Console.WriteLine("Volume of Box2 : {0}", volume);
Box3=Box1+Box2;
volume=Box3.getVolume();
Console.WriteLine("Volume of Box3 : {0}", volume);
Console.ReadKey();
}
}
}

```



OUTPUT :

```
[file:///c:/users/kalimuthu/documents/visual studio 2010/Projects/OPERA/OPERA/I  
Volume of Box1:210  
Volume of Box2:1560  
Volume of Box3:5400
```



EX.NO:6

Date:

SORTING THE NUMBERS IN ASCENDING ORDER

AIM:

To create a console application using operator overloading in C#.

PROGRAM:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

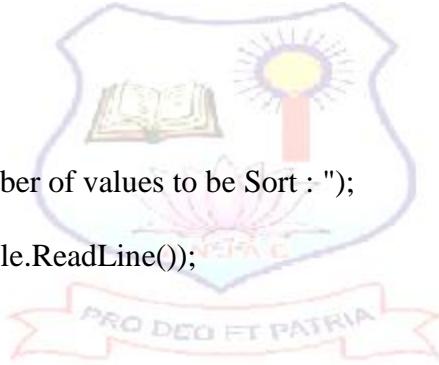
```
namespace sorting
{
    class Program
    {
        static void Main(string[] args)
        {
            int i;
            int[] a = new int[30];
            Console.Write("Enter the Number of values to be Sort :-");

            int n = Convert.ToInt16(Console.ReadLine());

            for (i = 1; i <= n; i++)
            {
                Console.Write("Enter the No " + i + ":");

                a[i] = Convert.ToInt16(Console.ReadLine());
            }

            for (i = 1; i <= n; i++)
            {
                for (int j = 1; j <= n - 1; j++)
                {
                    if (a[j] > a[j + 1])
                    {
                        int temp = a[j];
                        a[j] = a[j + 1];
                        a[j + 1] = temp;
                    }
                }
            }
        }
}
```



```
Console.WriteLine("Ascending Sort : ");
for (i = 1; i<= n; i++)
{
    Console.WriteLine(a[i]+" ");
}
Console.ReadKey();
}
}
```



OUTPUT:

```
file:///c:/users/lenovo/documents/visual studio 2010/Projects/sorting/sorting/bin/Debug/sorting.EXE
Enter the Number of values to be Sort : 5
Enter the No 1:89
Enter the No 2:67
Enter the No 3:34
Enter the No 4:12
Enter the No 5:20
Ascending Sort : 12 20 34 67 89
```



EX.NO:7

Date: **DISPLAY THE MAXIMUM & MINIMUM NUMBERS IN ARRAY**

AIM:

To create a console application using operator overloading in C#.

PROGRAM:

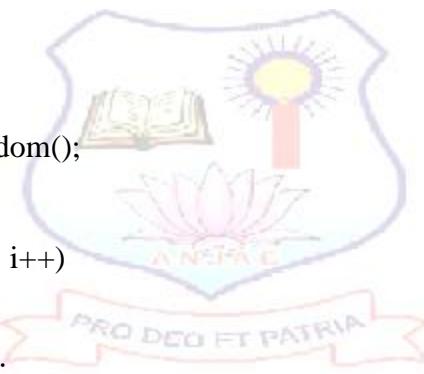
```
using System;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Text;

namespace max_and_min
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] numbers = new int[10];
            Random rnd = new Random();
            int min, max;

            for (int i = 0; i < numbers.Length; i++)
            {
                numbers[i] = rnd.Next(1, 100);
                Console.WriteLine(numbers[i]);
            }

            min = numbers[0];
            max = numbers[0];
            for (int i = 1; i < numbers.Length; i++)
            {
                if (min > numbers[i])
                    min = numbers[i];
                if (max < numbers[i])
                    max = numbers[i];
            }

            Console.WriteLine("=====");
            Console.WriteLine("The highest number in the array: " + max);
            Console.WriteLine("The lowest number in the array: " + min);
            Console.ReadKey();
        }
    }
}
```



OUTPUT:

```
file:///c/users/lenovo/documents/visual studio 2010/Projects/max and min/max and min/bin/Debug/max and min.EXE
16
31
56
95
11
60
81
85
73
42
=====
The highest number in the array: 95
The lowest number in the array: 11
```



EX.NO:8

Date:

DISPLAY THE TABLES FROM 1 TO 10

AIM:

To create a console application using operator overloading in C#.

PROGRAM:

```
namespace tables
{
    class Program
    {
        static void Main(string[] args)
        {

            for (int j = 1; j <= 10; j++)
            {
                for (int i = 1; i <= 10; i++)
                {
                    Console.WriteLine("{0}*{1}={2}\t", i, j, (i * j));
                }
                Console.WriteLine();
            }
            Console.ReadKey();
        }
    }
}
```



OUTPUT:

```
file:///c:/users/lenovo/documents/visual studio 2010/Projects/tables/bin/Debug/tables.EXE
1*1=1 2*1=2 3*1=3 4*1=4 5*1=5 6*1=6 7*1=7 8*1=8 9*1=9 10*1=10
1*2=2 2*2=4 3*2=6 4*2=8 5*2=10 6*2=12 7*2=14 8*2=16 9*2=18 10*2=20
1*3=3 2*3=6 3*3=9 4*3=12 5*3=15 6*3=18 7*3=21 8*3=24 9*3=27 10*3=30
1*4=4 2*4=8 3*4=12 4*4=16 5*4=20 6*4=24 7*4=28 8*4=32 9*4=36 10*4=40
1*5=5 2*5=10 3*5=15 4*5=20 5*5=25 6*5=30 7*5=35 8*5=40 9*5=45 10*5=50
1*6=6 2*6=12 3*6=18 4*6=24 5*6=30 6*6=36 7*6=42 8*6=48 9*6=54 10*6=60
1*7=7 2*7=14 3*7=21 4*7=28 5*7=35 6*7=42 7*7=49 8*7=56 9*7=63 10*7=70
1*8=8 2*8=16 3*8=24 4*8=32 5*8=40 6*8=48 7*8=56 8*8=64 9*8=72 10*8=80
1*9=9 2*9=18 3*9=27 4*9=36 5*9=45 6*9=54 7*9=63 8*9=72 9*9=81 10*9=90
1*10=10 2*10=20 3*10=30 4*10=40 5*10=50 6*10=60 7*10=70 8*10=80 9*10=90 10*10=100
```



EX.NO:9

Date:

DISPLAY THE CALCULATOR

AIM:

To create a console application using operator overloading in C#.

PROGRAM:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

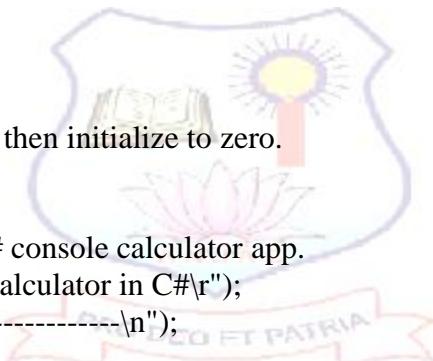
namespace CALCULATOR1
{
    class Program
    {
        static void Main(string[] args)
        {
            {
                // Declare variables and then initialize to zero.
                int num1 = 0; int num2 = 0;

                // Display title as the C# console calculator app.
                Console.WriteLine("Console Calculator in C#\r");
                Console.WriteLine("-----\n");

                // Ask the user to type the first number.
                Console.WriteLine("Type a number, and then press Enter");
                num1 = Convert.ToInt32(Console.ReadLine());

                // Ask the user to type the second number.
                Console.WriteLine("Type another number, and then press Enter");
                num2 = Convert.ToInt32(Console.ReadLine());

                // Ask the user to choose an option.
                Console.WriteLine("Choose an option from the following list:");
                Console.WriteLine("\ta - Add");
                Console.WriteLine("\ts - Subtract");
                Console.WriteLine("\tm - Multiply");
                Console.WriteLine("\td - Divide");
                Console.Write("Your option? ");
            }
        }
    }
}
```



```
// Use a switch statement to do the math.  
switch (Console.ReadLine())  
{  
    case "a":  
        Console.WriteLine("Your result: {num1} + {num2} = " + (num1 + num2));  
        break;  
    case "s":  
        Console.WriteLine("Your result: {num1} - {num2} = " + (num1 - num2));  
        break;  
    case "m":  
        Console.WriteLine("Your result: {num1} * {num2} = " + (num1 * num2));  
        break;  
    case "d":  
        Console.WriteLine("Your result: {num1} / {num2} = " + (num1 / num2));  
        break;  
}  
// Wait for the user to respond before closing.  
Console.Write("Press any key to close the Calculator console app...");  
Console.ReadKey();  
}  
}
```

OUTPUT:

```
file:///c:/users/lenovo/documents/visual studio 2010/Projects/CALCULATOR1/CALCULATOR1/bin/Debug/CALCULATOR1.EXE
Console Calculator in C#
-----
Type a number, and then press Enter
90
Type another number, and then press Enter
80
Choose an option from the following list:
    a - Add
    s - Subtract
    m - Multiply
    d - Divide
Your option? m
Your result: {num1} * {num2} = 7200
Press any key to close the Calculator console app...
```

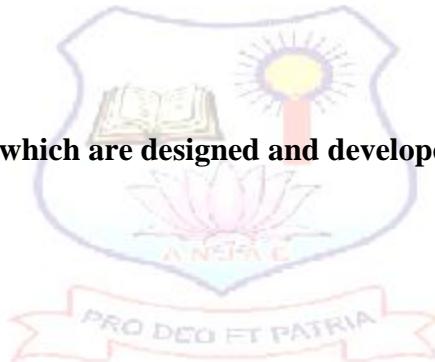


An Introduction to .NET Framework

.NET is a software framework which is designed and developed by Microsoft. The first version of .Net framework was 1.0 which came in the year 2002. In easy words, it is a virtual machine for compiling and executing programs written in different languages like C#, VB.Net etc.

It is used to develop Form-based applications, Web-based applications, and Web services. There is a variety of programming languages available on the .Net platform, VB.Net and C# being the most common ones are . It is used to build applications for Windows, phone, web etc. It provides a lot of functionalities and also supports industry standards.

.NET Framework supports more than 60 programming languages in which 11 programming languages are designed and developed by Microsoft. The remaining **Non-Microsoft Languages** which are supported by .NET Framework but not designed and developed by Microsoft.



11 Programming Languages which are designed and developed by Microsoft are:

- C#.NET
- VB.NET
- C++.NET
- J#.NET
- F#.NET
- JSCRIPT.NET
- WINDOWS POWERSHELL
- IRON RUBY
- IRON PYTHON
- C OMEGA
- ASML(Abstract State Machine Language)

Main Components of .NET Framework

Common Language Runtime(CLR):

CLR is the basic and Virtual Machine component of the .NET Framework. It is the run-time environment in the .NET Framework that runs the codes and helps in making the development process easier by providing the various services such as remoting, thread management, type-safety, memory management, robustness etc.. Basically, it is responsible for managing the execution of .NET programs regardless of any .NET programming language. It also helps in the management of code, as code that targets the runtime is known as the Managed Code and code doesn't target to runtime is known as Unmanaged code.

Framework Class Library(FCL):

It is the collection of reusable, object-oriented class libraries and methods etc that can be integrated with CLR. Also called the Assemblies. It is just like the header files in C/C++ and packages in the java. Installing .NET framework basically is the installation of CLR and FCL into the system.

.NET supports two kind of coding

1. Managed Code
2. Unmanaged Code

Managed Code

The resource, which is with in your application domain is, managed code. The resources that are within domain are faster.The code, which is developed in .NET framework, is known as managed code. This code is directly executed by CLR with help of managed code execution. Any language that is written in .NET Framework is managed code. Managed code uses CLR which in turns looks after your applications by managing memory, handling security, allowing cross - language debugging, and so on.

Unmanaged Code

The code, which is developed outside .NET Framework is known as unmanaged code. Applications that do not run under the control of the CLR are said to be unmanaged, and certain languages such as C++ can be used to write such applications, which, for example, access low - level functions of the operating system. Background compatibility with code of VB, ASP and COM are examples of unmanaged code. Unmanaged code can be unmanaged source code and unmanaged compile code. Unmanaged code is executed with help of wrapper classes.

Wrapper classes are of two types:

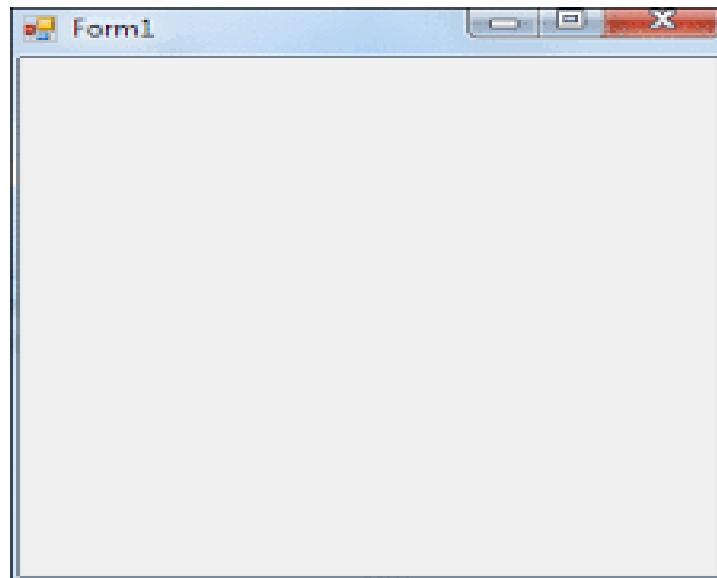
- CCW (COM Callable Wrapper)
- RCW (Runtime Callable Wrapper).

Wrapper is used to cover difference with the help of CCW and RCW.

FORMS

Let's start with creating a Window Forms Application by following the following steps in Microsoft Visual Studio - **File → New Project → Windows Forms Applications**

Finally, select OK, Microsoft Visual Studio creates your project and displays following window Form with a name **Form1**.



Form Properties

Following table lists down various important properties related to a form. These properties can be set or read during application execution. You can refer to Microsoft documentation for a complete list of properties associated with a Form control –

S.No	Properties	Description
1.	AcceptButton	The button that's automatically activated when you press Enter, no matter which control has the focus at the time. Usually the OK button on a form is set as AcceptButton for a form.
2.	CancelButton	The button that's automatically activated when you hit the Esc key. Usually, the Cancel button on a form is set as CancelButton for a form.
3.	AutoScale	This Boolean property determines whether the controls you place on the form are automatically scaled to the height of the current font. The default value of this property is True. This is a property of the form, but it affects the controls on the form.
4.	AutoScroll	This Boolean property indicates whether scroll bars will be automatically attached to the form if it is resized to a point that not all its controls are visible.
5.	AutoScrollMinSize	This property lets you specify the minimum size of the form, before the scroll bars are attached.
6.	AutoScrollPosition	The AutoScrollPosition is the number of pixels by which the two scroll bars were displaced from their initial locations.
7.	BackColor	Sets the form background color.

8.	BorderStyle	<p>The BorderStyle property determines the style of the form's border and the appearance of the form –</p> <ul style="list-style-type: none"> • None – Borderless window that can't be resized. • Sizable – This is default value and will be used for resizable window that's used for displaying regular forms. • Fixed3D – Window with a visible border, "raised" relative to the main area. In this case, windows can't be resized. • FixedDialog – A fixed window, used to create dialog boxes. • FixedSingle – A fixed window with a single line border. • FixedToolWindow – A fixed window with a Close button only. It looks like the toolbar displayed by the drawing and imaging applications. • SizableToolWindow – Same as the FixedToolWindow but resizable. In addition, its caption font is smaller than the usual.
9.	ControlBox	By default, this property is True and you can set it to False to hide the icon and disable the Control menu.
10.	Enabled	If True, allows the form to respond to mouse and keyboard events; if False, disables form.
11.	Font	This property specify font type, style, size
12.	HelpButton	Determines whether a Help button should be displayed in the caption box of the form.
13.	Height	This is the height of the Form in pixels.
14.	MinimizeBox	By default, this property is True and you can set it to False to hide the Minimize button on the title bar.
15.	MaximizeBox	By default, this property is True and you can set it to False to hide the Maximize button on the title bar.

16.	MinimumSize	This specifies the minimum height and width of the window you can minimize.
17.	MaximumSize	This specifies the maximum height and width of the window you maximize.
18.	Name	This is the actual name of the form.
19.	StartPosition	<p>This property determines the initial position of the form when it's first displayed. It will have any of the following values –</p> <ul style="list-style-type: none"> • CenterParent – The form is centered in the area of its parent form. • CenterScreen – The form is centered on the monitor. • Manual – The location and size of the form will determine its starting position. • WindowsDefaultBounds – The form is positioned at the default location and size determined by Windows. • WindowsDefaultLocation – The form is positioned at the Windows default location and has the dimensions you've set at design time.
20.	Text	The text, which will appear at the title bar of the form.
21.	Top, Left	These two properties set or return the coordinates of the form's top-left corner in pixels.
22.	TopMost	This property is a True/False value that lets you specify whether the form will remain on top of all other forms in your application. Its default property is False.
23.	Width	This is the width of the form in pixel.

Form Events

Following table lists down various important events related to a form. You can refer to Microsoft documentation for a complete list of events associated with forms control –

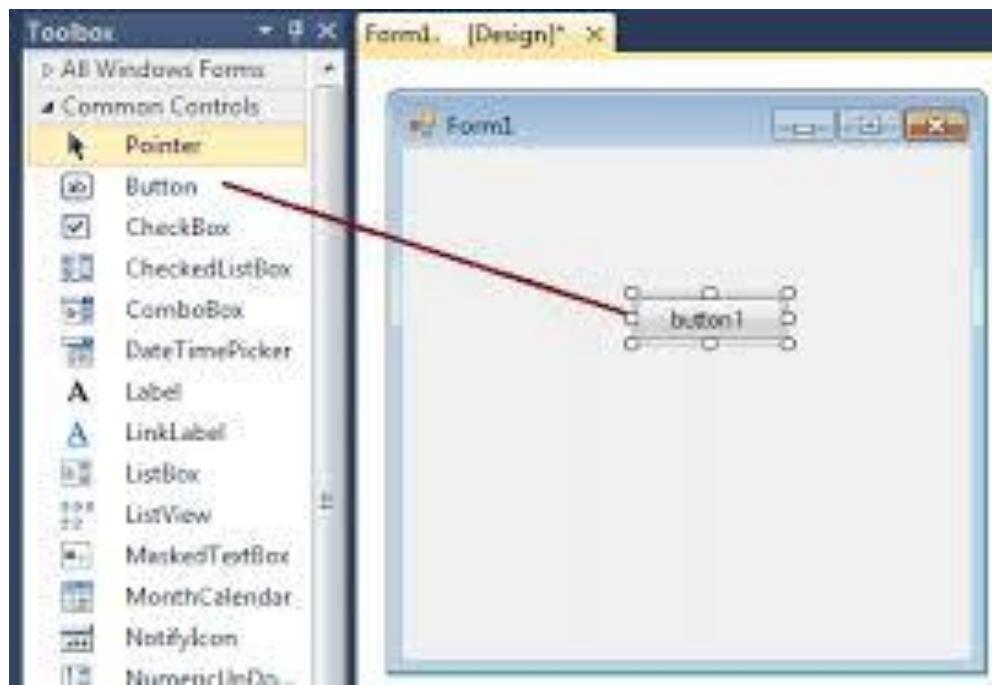
Sr.No.	Event	Description
1	Activated	Occurs when the form is activated in code or by the user.
2	Click	Occurs when the form is clicked.
3	Closed	Occurs before the form is closed.
4	Closing	Occurs when the form is closing.
5	DoubleClick	Occurs when the form control is double-clicked.
6	DragDrop	Occurs when a drag-and-drop operation is completed.
7	Enter	Occurs when the form is entered.
8	GotFocus	Occurs when the form control receives focus.
9	HelpButtonClicked	Occurs when the Help button is clicked.
10	KeyDown	Occurs when a key is pressed while the form has focus.
11	KeyPress	Occurs when a key is pressed while the form has focus.
12	KeyUp	Occurs when a key is released while the form has focus.
13	Load	Occurs before a form is displayed for the first time.
14	LostFocus	Occurs when the form loses focus.
15	MouseDown	Occurs when the mouse pointer is over the form and a mouse button is pressed.

16	MouseEnter	Occurs when the mouse pointer enters the form.
17	MouseHover	Occurs when the mouse pointer rests on the form.
18	MouseLeave	Occurs when the mouse pointer leaves the form.
19	MouseMove	Occurs when the mouse pointer is moved over the form.
20	MouseUp	Occurs when the mouse pointer is over the form and a mouse button is released.
21	MouseWheel	Occurs when the mouse wheel moves while the control has focus.
22	Move	Occurs when the form is moved.
23	Resize	Occurs when the control is resized.
24	Scroll	Occurs when the user or code scrolls through the client area.
25	Shown	Occurs whenever the form is first displayed.
26	VisibleChanged	Occurs when the Visible property value changes.

Button

The Button control represents a standard Windows button. It is generally used to generate a Click event by providing a handler for the Click event.

Let's create a label by dragging a Button control from the Toolbox ad dropping it on the form.



Properties of the Button Control

The following are some of the commonly used properties of the Button control

Sr.No.	Property & Description
1	AutoSizeMode Gets or sets the mode by which the Button automatically resizes itself.
2	BackColor Gets or sets the background color of the control.
3	BackgroundImage Gets or sets the background image displayed in the control.
4	DialogResult Gets or sets a value that is returned to the parent form when the button is clicked. This is used while creating dialog boxes.
5	ForeColor Gets or sets the foreground color of the control.

6	Image Gets or sets the image that is displayed on a button control.
7	Location Gets or sets the coordinates of the upper-left corner of the control relative to the upper-left corner of its container.
8	TabIndex Gets or sets the tab order of the control within its container.
9	Text Gets or sets the text associated with this control.

Methods of the Button Control

The following are some of the commonly used methods of the Button control

Sr.No.	Method Name & Description
1.	GetPreferredSize Retrieves the size of a rectangular area into which a control can be fitted.
2.	NotifyDefault Notifies the Button whether it is the default button
3.	Select Activates the control.
4.	ToString Returns a String containing the name of the Component, if any. This method should not be overridden.

Events of the Button Control

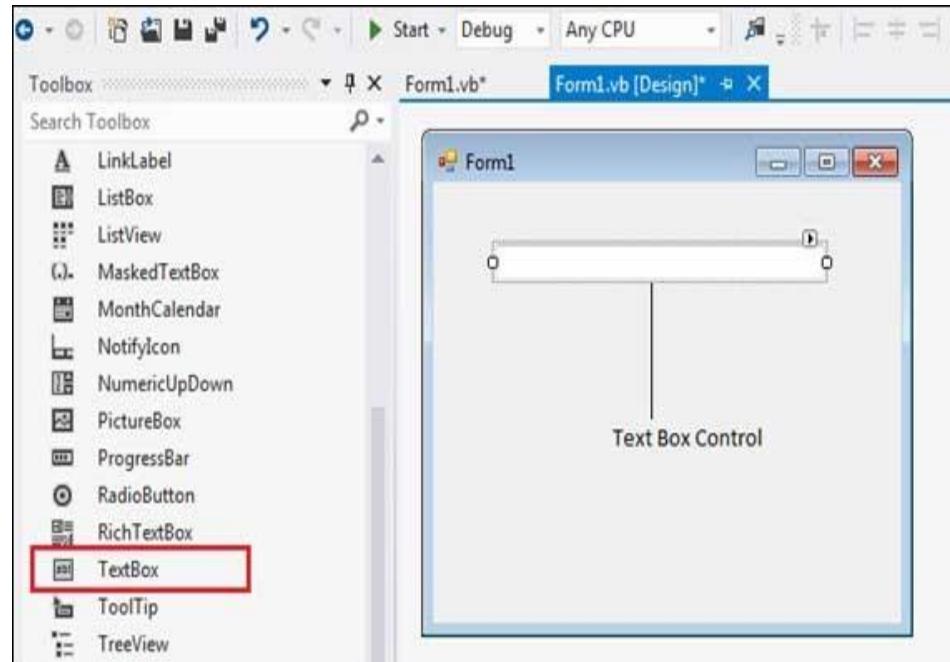
The following are some of the commonly used events of the Button control

Sr.No.	Event & Description
1.	Click Occurs when the control is clicked.
2.	DoubleClick Occurs when the user double-clicks the Button control.
3.	GotFocus Occurs when the control receives focus.
4.	TabIndexChanged Occurs when the TabIndex property value changes.
5.	TextChanged Occurs when the Text property value changes.
6.	Validated Occurs when the control is finished validating.

Text Box

Text box controls allow entering text on a form at runtime. By default, it takes a single line of text, however, you can make it accept multiple texts and even add scroll bars to it.

Let's create a text box by dragging a Text Box control from the Toolbox and dropping it on the form.



The Properties of the TextBox Control

The following are some of the commonly used properties of the TextBox control –

Sr.No.	Property & Description
1.	AcceptsReturn Gets or sets a value indicating whether pressing ENTER in a multiline TextBox control creates a new line of text in the control or activates the default button for the form.
2.	AutoCompleteCustomSource Gets or sets a custom System.Collections.Specialized.StringCollection to use when the AutoCompleteSourceproperty is set to CustomSource.
3.	AutoCompleteMode Gets or sets an option that controls how automatic completion works for the TextBox.

4.	AutoCompleteSource Gets or sets a value specifying the source of complete strings used for automatic completion.
5.	CharacterCasing Gets or sets whether the TextBox control modifies the case of characters as they are typed.
6.	Font Gets or sets the font of the text displayed by the control.
7.	FontHeight Gets or sets the height of the font of the control.
8.	ForeColor Gets or sets the foreground color of the control.
9.	Lines Gets or sets the lines of text in a text box control.
10.	Multiline Gets or sets a value indicating whether this is a multiline TextBox control.
11.	PasswordChar Gets or sets the character used to mask characters of a password in a single-line TextBox control.
12.	ReadOnly Gets or sets a value indicating whether text in the text box is read-only.
13.	ScrollBars Gets or sets which scroll bars should appear in a multiline TextBox control. This property has values – <ul style="list-style-type: none"> • None • Horizontal • Vertical • Both

14.	TabIndex Gets or sets the tab order of the control within its container.
15.	Text Gets or sets the current text in the TextBox.
16.	 TextAlign Gets or sets how text is aligned in a TextBox control. This property has values – <ul style="list-style-type: none"> • Left • Right • Center
17.	TextLength Gets the length of text in the control.
18.	WordWrap Indicates whether a multiline text box control automatically wraps words to the beginning of the next line when necessary.

Methods of the TextBox Control

The following are some of the commonly used methods of the TextBox control

Sr.No.	Method Name & Description
1.	AppendText Appends text to the current text of a text box.
2.	Clear Clears all text from the text box control.
3.	Copy Copies the current selection in the text box to the Clipboard .

4.	Cut Moves the current selection in the text box to the Clipboard .
5.	Paste Replaces the current selection in the text box with the contents of the Clipboard .
6.	Paste(String) Sets the selected text to the specified text without clearing the undo buffer.
7.	ResetText Resets the Text property to its default value.
8.	ToString Returns a string that represents the TextBoxBase control.
9.	Undo Undoes the last edit operation in the text box.

Events of the Text Box Control

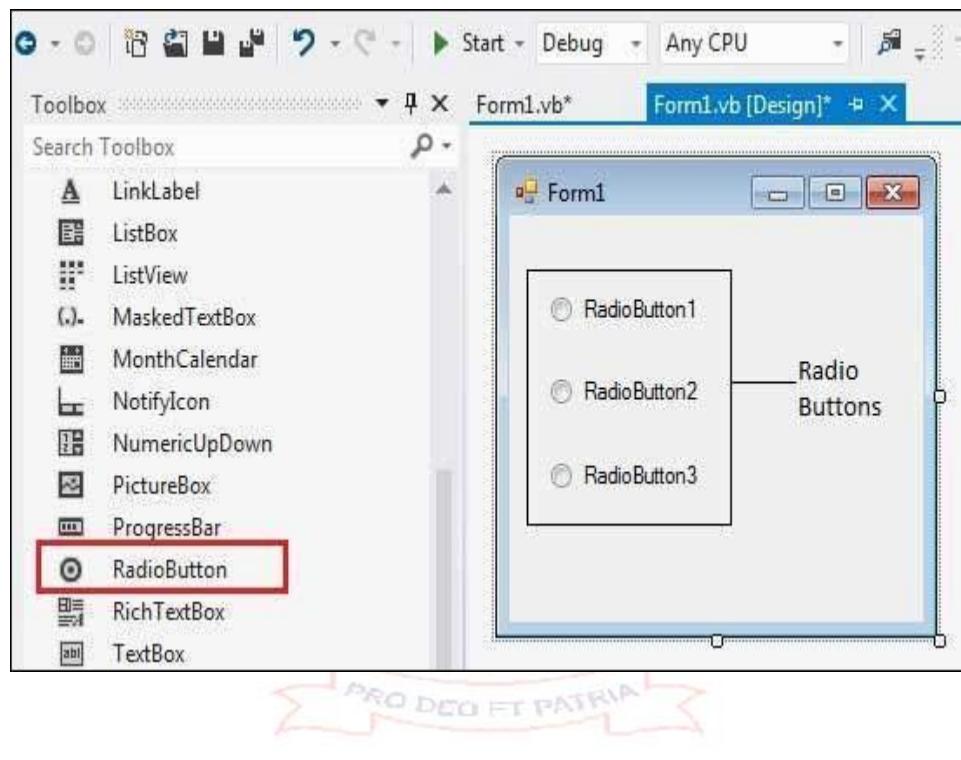
The following are some of the commonly used events of the Text control –

Sr.No.	Event & Description
1.	Click Occurs when the control is clicked.
2.	DoubleClick Occurs when the control is double-clicked.
3.	TextAlignChanged Occurs when the TextAlign property value changes.

RadioButton

The RadioButton control is used to provide a set of mutually exclusive options. The user can select one radio button in a group. If you need to place more than one group of radio buttons in the same form, you should place them in different container controls like a GroupBox control.

Let's create three radio buttons by dragging RadioButton controls from the Toolbox and dropping on the form.



The Checked property of the radio button is used to set the state of a radio button. You can display text, image or both on radio button control. You can also change the appearance of the radio button control by using the Appearance property.

Properties of the RadioButton Control

The following are some of the commonly used properties of the RadioButton control

Sr.No.	Property & Description
1.	Appearance Gets or sets a value determining the appearance of the radio button.

2.	AutoCheck Gets or sets a value indicating whether the Checked value and the appearance of the control automatically change when the control is clicked.
3.	CheckAlign Gets or sets the location of the check box portion of the radio button.
4.	Checked Gets or sets a value indicating whether the control is checked.
5.	Text Gets or sets the caption for a radio button.
6.	TabStop Gets or sets a value indicating whether a user can give focus to the RadioButton control using the TAB key.

Methods of the RadioButton Control

The following are some of the commonly used methods of the RadioButton control

Sr.No.	Method Name & Description
1.	PerformClick Generates a Click event for the control, simulating a click by a user.

Events of the RadioButton Control

The following are some of the commonly used events of the RadioButton control

Sr.No	Event & Description
1.	AppearanceChanged Occurs when the value of the Appearance property of the RadioButton control is changed.

2.

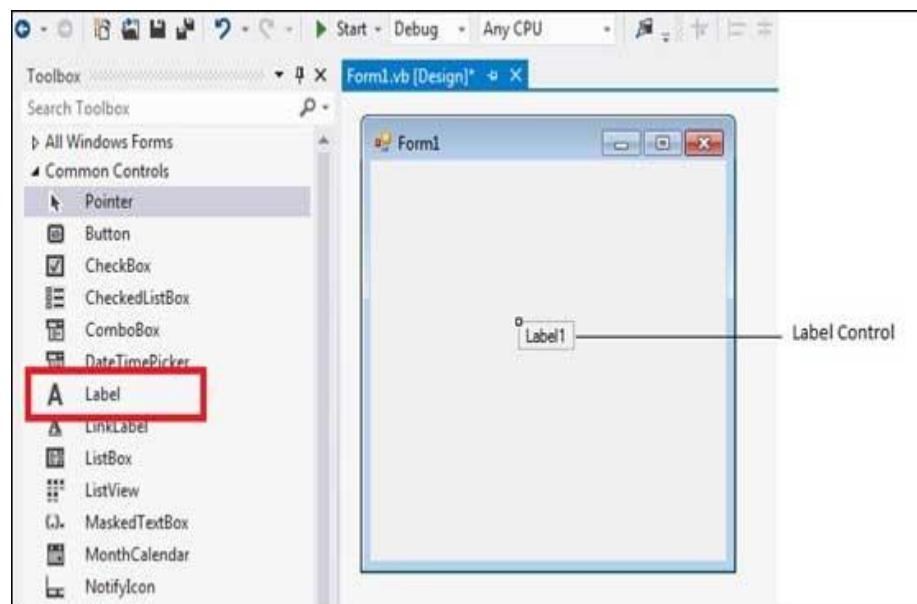
CheckedChanged

Occurs when the value of the Checked property of the RadioButton control is changed.

Label

The Label control represents a standard Windows label. It is generally used to display some informative text on the GUI which is not changed during runtime.

Let's create a label by dragging a Label control from the Toolbox and dropping it on the form.



Properties of the Label Control

The following are some of the commonly used properties of the Label control

Sr.No.	Property & Description
1.	Autosize Gets or sets a value specifying if the control should be automatically resized to display all its contents.
2.	BorderStyle Gets or sets the border style for the control.

3.	FlatStyle Gets or sets the flat style appearance of the Label control
4.	Font Gets or sets the font of the text displayed by the control.
5.	FontHeight Gets or sets the height of the font of the control.
6.	ForeColor Gets or sets the foreground color of the control.
7.	PreferredSize Gets the preferred height of the control.
8.	PreferredWidth Gets the preferred width of the control.
9.	TabStop Gets or sets a value indicating whether the user can tab to the Label. This property is not used by this class.
10.	Text Gets or sets the text associated with this control.
11.	.TextAlign Gets or sets the alignment of text in the label.

Methods of the Label Control

The following are some of the commonly used methods of the Label control

Sr.No.	Method Name & Description
1.	GetPreferredSize Retrieves the size of a rectangular area into which a control can be fitted.

2.	Refresh Forces the control to invalidate its client area and immediately redraw itself and any child controls.
3.	Select Activates the control.
4.	Show Displays the control to the user.
5.	ToString Returns a String that contains the name of the control.

Events of the Label Control

The following are some of the commonly used events of the Label control –

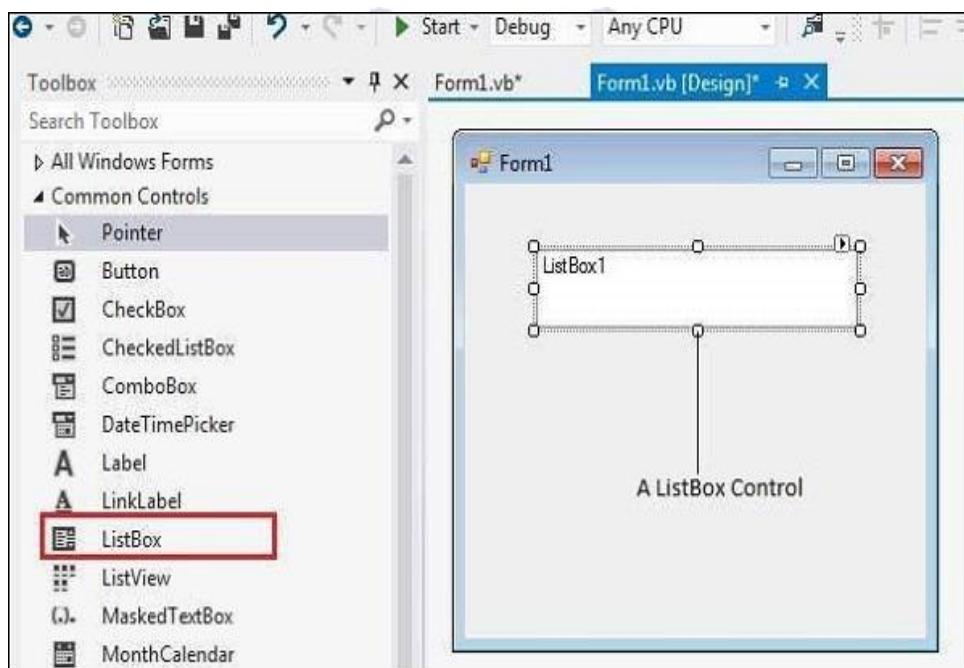
Sr.No.	Event & Description
1.	AutoSizeChanged Occurs when the value of the AutoSize property changes.
2.	Click Occurs when the control is clicked.
3.	DoubleClick Occurs when the control is double-clicked.
4.	GotFocus Occurs when the control receives focus.
5.	Leave Occurs when the input focus leaves the control.
6.	LostFocus Occurs when the control loses focus.

7.	TabIndexChanged Occurs when the TabIndex property value changes.
8.	TabStopChanged Occurs when the TabStop property changes.
9.	TextChanged Occurs when the Text property value changes.

List Box

The ListBox represents a Windows control to display a list of items to a user. A user can select an item from the list. It allows the programmer to add items at design time by using the properties window or at the runtime.

Let's create a list box by dragging a ListBox control from the Toolbox and dropping it on the form.



You can populate the list box items either from the properties window or at runtime. To add items to a ListBox, select the ListBox control and get to the properties window, for the properties of this control. Click the ellipses (...) button next to the Items property. This opens the String Collection Editor dialog box, where you can enter the values one at a line.

Properties of the ListBox Control

The following are some of the commonly used properties of the ListBox control –

Sr.No.	Property & Description
1.	AllowSelection Gets a value indicating whether the ListBox currently enables selection of list items.
2.	BorderStyle Gets or sets the type of border drawn around the list box.
3.	ColumnWidth Gets or sets the width of columns in a multicolumn list box.
4.	HorizontalExtent Gets or sets the horizontal scrolling area of a list box.
5.	HorizontalScrollBar Gets or sets the value indicating whether a horizontal scrollbar is displayed in the list box.
6.	ItemHeight Gets or sets the height of an item in the list box.
7.	Items Gets the items of the list box.
8.	MultiColumn Gets or sets a value indicating whether the list box supports multiple columns.
9.	ScrollAlwaysVisible Gets or sets a value indicating whether the vertical scroll bar is shown at all times.
10.	SelectedIndex Gets or sets the zero-based index of the currently selected item in a list box.

11.	SelectedIndices Gets a collection that contains the zero-based indexes of all currently selected items in the list box.
12.	SelectedItem Gets or sets the currently selected item in the list box.
13.	SelectedItems Gets a collection containing the currently selected items in the list box.
14.	SelectedValue Gets or sets the value of the member property specified by the ValueMember property.
15.	SelectionMode Gets or sets the method in which items are selected in the list box. This property has values – <ul style="list-style-type: none"> • None • One • MultiSimple • MultiExtended 
16.	Sorted Gets or sets a value indicating whether the items in the list box are sorted alphabetically.
17.	Text Gets or searches for the text of the currently selected item in the list box.
18.	TopIndex Gets or sets the index of the first visible item of a list box.

Methods of the ListBox Control

The following are some of the commonly used methods of the ListBox control

Sr.No.	Method Name & Description
1.	BeginUpdate Prevents the control from drawing until the EndUpdate method is called, while items are added to the ListBox one at a time.
2.	ClearSelected Unselects all items in the ListBox.
3.	EndUpdate Resumes drawing of a list box after it was turned off by the BeginUpdate method.
4.	FindString Finds the first item in the ListBox that starts with the string specified as an argument.
5.	FindStringExact Finds the first item in the ListBox that exactly matches the specified string.
6.	GetSelected Returns a value indicating whether the specified item is selected.
7.	SetSelected Selects or clears the selection for the specified item in a ListBox.
8.	OnSelectedIndexChanged Raises the SelectedIndexChanged event.
9.	OnSelectedValueChanged Raises the SelectedValueChanged event.

Events of the ListBox Control

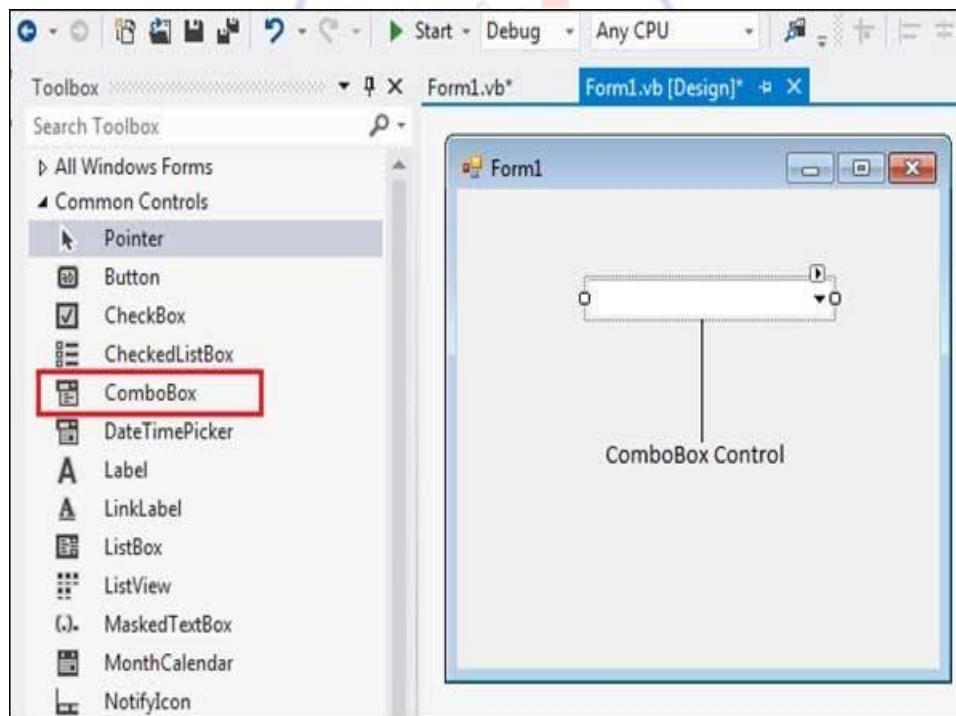
The following are some of the commonly used events of the ListBox control

Sr.No.	Event & Description
1.	Click Occurs when a list box is selected.
2.	SelectedIndexChanged Occurs when the SelectedIndex property of a list box is changed.

Combo Box

The ComboBox control is used to display a drop-down list of various items. It is a combination of a text box in which the user enters an item and a drop-down list from which the user selects an item.

Let's create a combo box by dragging a ComboBox control from the Toolbox and dropping it on the form.



You can populate the list box items either from the properties window or at runtime. To add items to a ComboBox, select the ComboBox control and go to the properties window for the properties of this control. Click the ellipses (...) button next to the Items property. This opens the String Collection Editor dialog box, where you can enter the values one at a line.

Properties of the ComboBox Control

The following are some of the commonly used properties of the ComboBox control

Sr.No.	Property & Description
1.	AllowSelection Gets a value indicating whether the list enables selection of list items.
2.	AutoCompleteCustomSource Gets or sets a custom System.Collections.Specialized.StringCollection to use when the AutoCompleteSourceproperty is set to CustomSource.
3.	AutoCompleteMode Gets or sets an option that controls how automatic completion works for the ComboBox.
4.	AutoCompleteSource Gets or sets a value specifying the source of complete strings used for automatic completion.
5.	DataBindings Gets the data bindings for the control.
6.	DataManager Gets the CurrencyManager associated with this control.
7.	DataSource Gets or sets the data source for this ComboBox.
8.	DropDownHeight Gets or sets the height in pixels of the drop-down portion of the ComboBox.
9.	DropDownStyle Gets or sets a value specifying the style of the combo box.

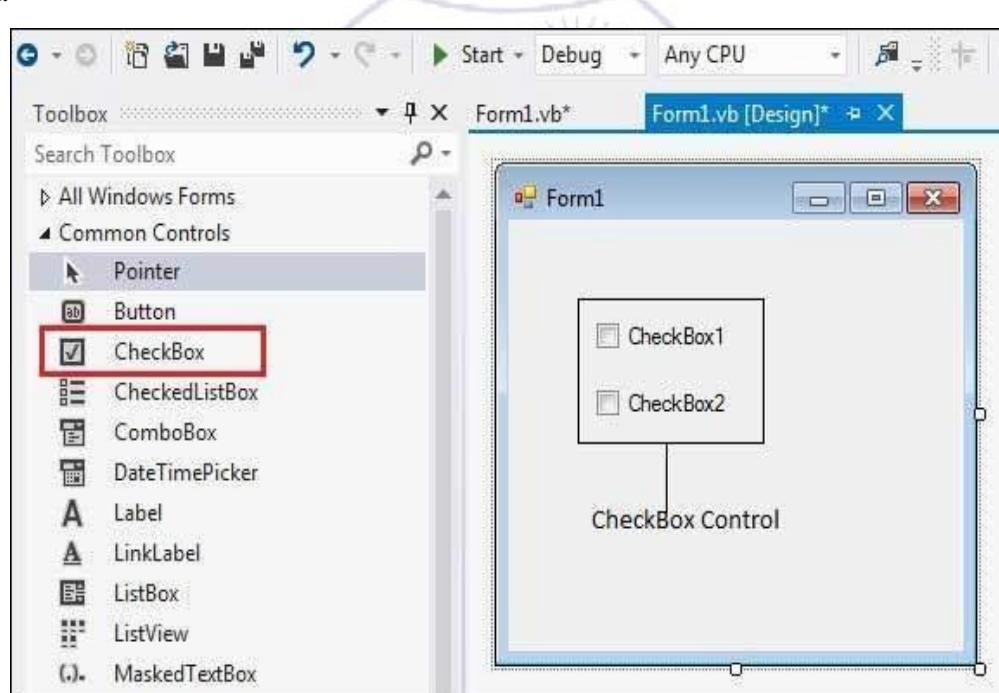
10.	DropDownWidth Gets or sets the width of the drop-down portion of a combo box.
11.	DroppedDown Gets or sets a value indicating whether the combo box is displaying its drop-down portion.
12.	FlatStyle Gets or sets the appearance of the ComboBox.
13.	ItemHeight Gets or sets the height of an item in the combo box.
14.	Items Gets an object representing the collection of the items contained in this ComboBox.
15.	MaxDropDownItems Gets or sets the maximum number of items to be displayed in the drop-down part of the combo box.
16.	MaxLength Gets or sets the maximum number of characters a user can enter in the editable area of the combo box.
17.	SelectedIndex Gets or sets the index specifying the currently selected item.
18.	SelectedItem Gets or sets currently selected item in the ComboBox.
19.	SelectedText Gets or sets the text that is selected in the editable portion of a ComboBox.
20.	SelectedValue Gets or sets the value of the member property specified by the ValueMember property.
21.	SelectionLength Gets or sets the number of characters selected in the editable portion of the combo box.

22.	SelectionStart Gets or sets the starting index of text selected in the combo box.
23.	Sorted Gets or sets a value indicating whether the items in the combo box are sorted.
24.	Text Gets or sets the text associated with this control.

Check Box

The CheckBox control allows the user to set true/false or yes/no type options. The user can select or deselect it. When a check box is selected it has the value True, and when it is cleared, it holds the value False.

Let's create two check boxes by dragging CheckBox controls from the Toolbox and dropping on the form.



The CheckBox control has three states, **checked**, **unchecked** and **indeterminate**. In the indeterminate state, the check box is grayed out. To enable the indeterminate state, the ThreeState property of the check box is set to be **True**.

Properties of the CheckBox Control

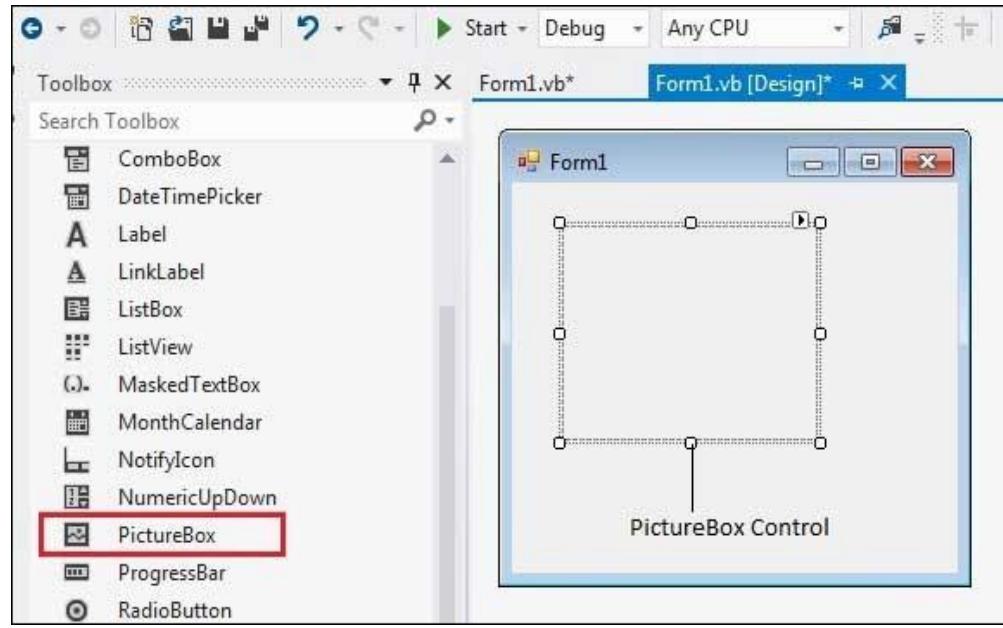
The following are some of the commonly used properties of the CheckBox control .

Sr.No.	Property & Description
1.	Appearance Gets or sets a value determining the appearance of the check box.
2.	AutoCheck Gets or sets a value indicating whether the Checked or CheckedState value and the appearance of the control automatically change when the check box is selected.
3.	CheckAlign Gets or sets the horizontal and vertical alignment of the check mark on the check box.
4.	Checked Gets or sets a value indicating whether the check box is selected.
5.	CheckState Gets or sets the state of a check box.
6.	Text Gets or sets the caption of a check box.
7.	ThreeState Gets or sets a value indicating whether or not a check box should allow three check states rather than two.

Picture Box

The PictureBox control is used for displaying images on the form. The Image property of the control allows you to set an image both at design time or at run time.

Let's create a picture box by dragging a PictureBox control from the Toolbox and dropping it on the form.



Properties of the PictureBox Control

The following are some of the commonly used properties of the PictureBox control –

Sr.No.	Property & Description
1.	AllowDrop Specifies whether the picture box accepts data that a user drags on it.
2.	ErrorImage Gets or specifies an image to be displayed when an error occurs during the image-loading process or if the image load is cancelled.
3.	Image Gets or sets the image that is displayed in the control.
4.	ImageLocation Gets or sets the path or the URL for the image displayed in the control.
5.	InitialImage Gets or sets the image displayed in the control when the main image is loaded.
6.	SizeMode Determines the size of the image to be displayed in the control. This property

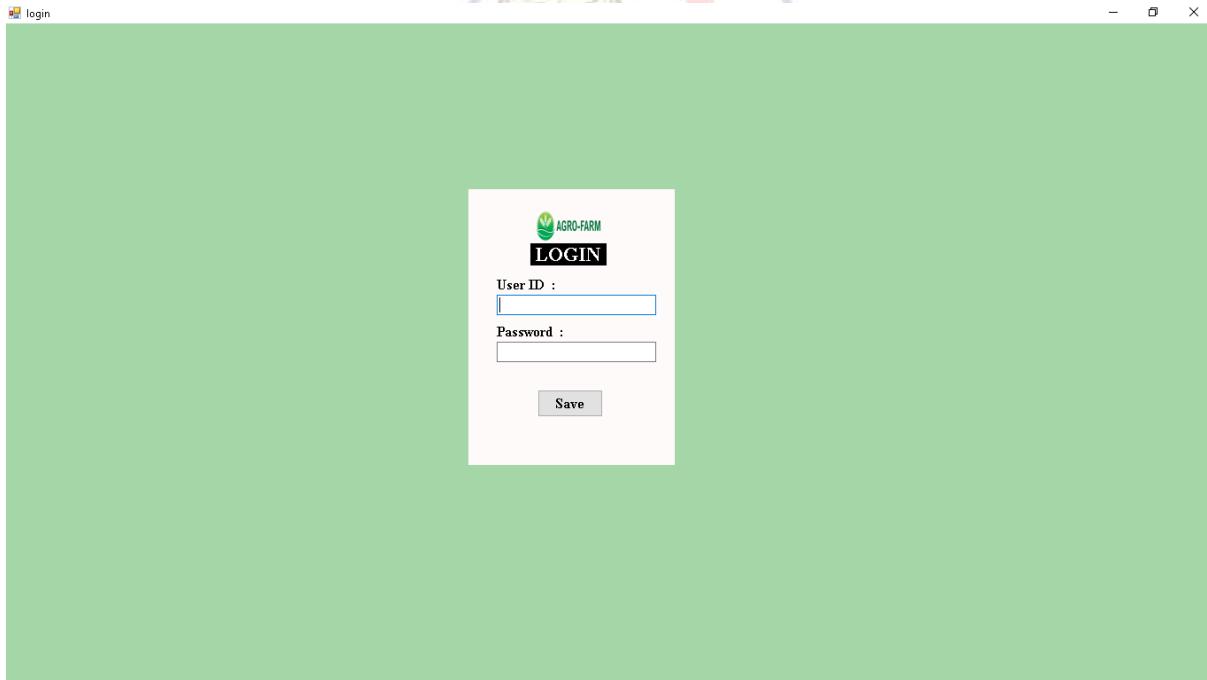
	<p>takes its value from the PictureBoxSizeMode enumeration, which has values –</p> <ul style="list-style-type: none"> • Normal – the upper left corner of the image is placed at upper left part of the picture box • StrechImage – allows stretching of the image • AutoSize – allows resizing the picture box to the size of the image • CenterImage – allows centering the image in the picture box • Zoom – allows increasing or decreasing the image size to maintain the size ratio.
7.	<p>TabIndex</p> <p>Gets or sets the tab index value.</p>
8.	<p>TabStop</p> <p>Specifies whether the user will be able to focus on the picture box by using the TAB key.</p>
9.	<p>Text</p> <p>Gets or sets the text for the picture box.</p>
10.	<p>WaitOnLoad</p> <p>Specifies whether or not an image is loaded synchronously.</p>

Ex. No :10**Date :** **Agriculture Management System****Aim :**

To create a windows application for the maintenance of the agriculture field.

Program :**Login:**

```
If TextBox1.Text = "admin" And TextBox2.Text = "admin" Then  
Form1.Show()  
    TextBox1.Text = ""  
    TextBox2.Text = ""  
ElseIf TextBox1.Text = "report" And TextBox2.Text = "report" Then  
Form2.Show()  
    TextBox1.Text = ""  
    TextBox2.Text = ""  
Else  
    MsgBox("Login Failed !!!")  
End If
```



Employee Work List :

AGRICULTURE MANAGEMENT SYSTEM

 AGRO-FARM

Report

- Vehicle/Equipment Type Report
- Vehicle/Equipment Details
- Vehicle/Equipment Price Details
- Vehicle/Equipment List
- Employee Details
- Employee Wage Details
- Employee Personal Details
- Employee Work Details
- Soil Details
- Land Details
- Income Details
- Expense Details

Employee Work List

	ID	code	name	work
▶	1	khjh	hkjh	harvest , inspect ...
*				

Copyright © 2021 Anjana Infotech. All rights reserved.

Soil Details List :

AGRICULTURE MANAGEMENT SYSTEM

 AGRO-FARM

Report

- Vehicle/Equipment Type Report
- Vehicle/Equipment Details
- Vehicle/Equipment Price Details
- Vehicle/Equipment List
- Employee Details
- Employee Wage Details
- Employee Personal Details
- Employee Work Details
- Soil Details
- Land Details
- Income Details
- Expense Details

Soil Details List

	ID	code	surveyno	details	landtype	totalsize	name
▶	1	kh	hkjh	jhkj	fertile soil	fertilize , remove ...	fertile soil
*							

Copyright © 2021 Anjana Infotech. All rights reserved.

Employee Personal Detail List :

AGRICULTURE MANAGEMENT SYSTEM

 AGRO-FARM

Report
Vehicle/Equipment Type Report
Vehicle/Equipment Details
Vehicle/Equipment Price Details
Vehicle/Equipment List
Employee Details
Employee Wage Details
Employee Personal Details
Employee Work Details
Soil Details
Land Details
Income Details
Expense Details

Employee Personal Details List

	ID	code	name	address	mobile
▶	1	khjh	hkjh	jk	hj
*					

Copyright © 2021 Anjana Infotech. All rights reserved.

Employee Wage List :

AGRICULTURE MANAGEMENT SYSTEM

 AGRO-FARM

Report
Vehicle/Equipment Type Report
Vehicle/Equipment Details
Vehicle/Equipment Price Details
Vehicle/Equipment List
Employee Details
Employee Wage Details
Employee Personal Details
Employee Work Details
Soil Details
Land Details
Income Details
Expense Details

Employee Wage List

	ID	code	name	wage
▶	1	khjh	hkjh	kghkj
*				

Copyright © 2021 Anjana Infotech. All rights reserved.

Employee Details List :

AGRICULTURE MANAGEMENT SYSTEM

 AGRO-FARM

ID	code	name	address	mobile	work	wage
1	khjh	hjhj	jk	hi	harvest, inspect...	kjhkj
*						

Report

Vehicle/Equipment Type Report

Vehicle/Equipment Details

Vehicle/Equipment Price Details

Vehicle/Equipment List

Employee Details

Employee Wage Details

Employee Personal Details

Employee Work Details

Soil Details

Land Details

Income Details

Expense Details

Employee Details List

Submit

Copyright © 2021 Anjana Infotech. All rights reserved.

Vehicle / Equipment List

AGRICULTURE MANAGEMENT SYSTEM

 AGRO-FARM

ID	name
1	jhjh
*	

Menu

Report

Vehicle/Equipment Details

Vehicle/Equipment Price List

Vehicle/Equipment Type

Vehicle/Equipment Name List

Vehicle/Equipment Type

Vehicle/Equipment Details

Vehicle / Equipment List

Submit

Copyright © 2021 Anjana Infotech. All rights reserved.

Vehicle / Equipment type Details:

AGRICULTURE MANAGEMENT SYSTEM

 AGRO-FARM

Menu

Report

Vehicle/Equipment Details

Vehicle/Equipment Price List

Vehicle/Equipment Type

Vehicle/Equipment Name List

Vehicle/Equipment Type

Vehicle/Equipment Details

Vehicle / Equipment Type Details

Submit

ID	name	type
1	hh	h
*		

< >

Copyright © 2021 Anjana Infotech. All rights reserved.

Vehicle / EquipmentPrice Details :

AGRICULTURE MANAGEMENT SYSTEM

 AGRO-FARM

Menu

Report

Vehicle/Equipment Details

Vehicle/Equipment Price List

Vehicle/Equipment Type

Vehicle/Equipment Name List

Vehicle/Equipment Type

Vehicle/Equipment Details

Vehicle / Equipment Price Details

Submit

ID	name	type	price
1	hh	h	hh
*			

< >

Copyright © 2021 Anjana Infotech. All rights reserved.

Vehicle / Equipment Details:

AGRICULTURE MANAGEMENT SYSTEM



AGRO-FARM

Menu	Vehicle / Equipment Details				
Report	<input type="button" value="Submit"/>				
Vehicle/Equipment Details					
Vehicle/Equipment Price List	ID	name	type	price	details
Vehicle/Equipment Type	▶ 1	jhjh	hj	hjhj	khkh
Vehicle/Equipment Name List	*				
Vehicle/Equipment Type					
Vehicle/Equipment Details					

Copyright © 2021 Anjana Infotech. All rights reserved.

Income Details :

```
If (TextBox1.Text = "" And TextBox2.Text = "") Then
    MsgBox("Enter all the details!!")
```

```
Else
```

```
rec.Open("select * from income", con, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)
rec.AddNew()
```

```
rec(1).Value = TextBox1.Text
```

```
rec(2).Value = DateTimePicker1.Text
```

```
rec(3).Value = TextBox2.Text
```

```
rec(4).Value = TextBox4.Text
```

```
rec(5).Value = TextBox5.Text
```

```
rec(6).Value = TextBox3.Text
```

```
rec.Update()
```

```
MsgBox("Income Details Added!!")
```

```
rec.Close()
```

```
End If
```

```
Me.Close()
```



Menu
Vehicle/Equipment Type
Vehicle/Equipment Details
Employment Details
Land Details
Crop Details
Expenses
Income

Report

Income Details

Voucher No. :	<input type="text"/>
Date :	10 November 2021 <input type="button" value=""/>
Details :	<input type="text"/>
Received From :	<input type="text"/>
Amount :	<input type="text"/>
Remark :	<input type="text"/>

Copyright © 2021 Anjana Infotech. All rights reserved.

Expense Details :

If (TextBox1.Text = "" And TextBox2.Text = "") Then
 MsgBox("Enter all the details!!")
 Else
 rec.Open("select * from expense", con, ADODB.CursorTypeEnum.adOpenDynamic,
 ADODB.LockTypeEnum.adLockOptimistic)
 rec.AddNew()
 rec(1).Value = TextBox1.Text
 rec(2).Value = Datepicker1.Text
 rec(3).Value = TextBox2.Text
 rec(4).Value = TextBox5.Text
 rec(5).Value = TextBox3.Text
 rec.Update()
 MsgBox("Expense Details Added!!")
 rec.Close()
 End If
 Me.Close()

**Menu**

Vehicle/Equipment Type

Vehicle/Equipment Details

Employment Details

Land Details

Crop Details

Expenses

Income

Report**Expenses Details**Voucher No. : Date :

Details :

Amount : Remark :

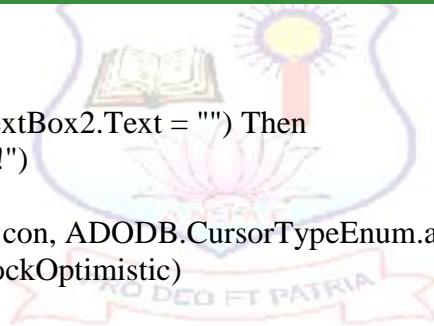
Copyright © 2021 Anjana Infotech. All rights reserved.

Crop Details :

```

If (TextBox1.Text = "" And TextBox2.Text = "") Then
MsgBox("Enter all the details!!")
Else
rec.Open("select * from crop", con, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)
rec.AddNew()
rec(1).Value = TextBox1.Text
rec(2).Value = TextBox4.Text
rec(3).Value = TextBox3.Text
rec(4).Value = TextBox2.Text
rec.Update()
MsgBox("Crop Details Added!!")
rec.Close()
End If
Me.Close()

```





Menu

Vehicle/Equipment Type

Vehicle/Equipment Details

Employment Details

Land Details

Crop Details

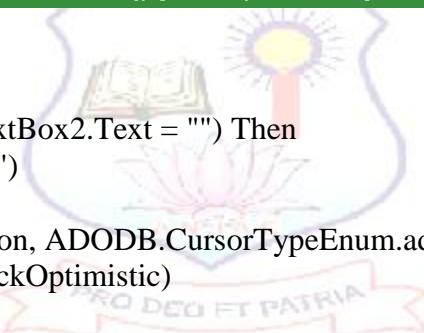
Expenses

Income

Report

Crop DetailsCrop Code : Crop Name : Season : Details :

Copyright © 2021 Anjana Infotech. All rights reserved.

**Land Details :**

```

If (TextBox1.Text = "" And TextBox2.Text = "") Then
    MsgBox("Enter all the details!!")
Else
    rec.Open("select * from soil", con, ADODB.CursorTypeEnum.adOpenDynamic,
        ADODB.LockTypeEnum.adLockOptimistic)
    rec.AddNew()
    rec(1).Value = TextBox1.Text
    rec(2).Value = TextBox4.Text
    rec(3).Value = TextBox2.Text
    rec(4).Value = ComboBox2.Text
    rec(5).Value = TextBox3.Text
    rec(6).Value = ComboBox3.Text
    rec.Update()
    MsgBox("Land Details Added!!")
    rec.Close()
End If
Me.Close()

```

**Menu**

Vehicle/Equipment Type

Vehicle/Equipment Details

Employment Details

Land Details

Crop Details

Expenses

Income

Report**Land Details**Land Code : Survey No. : Details : Land Type : Total Size : Emp name : **Submit**

Copyright © 2021 Anjana Infotech. All rights reserved.

Employee Details :

If (TextBox1.Text = "" And TextBox2.Text = "") Then
 MsgBox("Enter all the details!!")
 Else
 rec.Open("select * from emp", con, ADODB.CursorTypeEnum.adOpenDynamic,
 ADODB.LockTypeEnum.adLockOptimistic)
 rec.AddNew()
 rec(1).Value = TextBox1.Text
 rec(2).Value = TextBox4.Text
 rec(3).Value = TextBox2.Text
 rec(4).Value = TextBox3.Text
 rec(5).Value = ComboBox1.Text
 rec(6).Value = TextBox5.Text
 rec.Update()
 MsgBox("Employee Details Added!!")
 rec.Close()
 End If
 Me.Close()



Menu

Vehicle/Equipment Type

Vehicle/Equipment Details

Employment Details

Land Details

Crop Details

Expenses

Income

Report

Employee DetailsEmployee Code : Employee Name :

Address :

Mobile :

Type Of Work :

Wages Per Day :

Copyright © 2021 Anjana Infotech. All rights reserved.

Veehicle / Equipment Details :

If (TextBox1.Text = "" And TextBox2.Text = "") Then

MsgBox("Enter all the details!!")

Else

rec.Open("select * from vedetails", con, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)

rec.AddNew()

rec(1).Value = TextBox1.Text

rec(2).Value = ComboBox1.Text

rec(3).Value = TextBox3.Text

rec(4).Value = TextBox2.Text

rec.Update()

MsgBox("Vehicle / Equipment Details Added!!")

rec.Close()

End If

Me.Close()

AGRICULTURE MANAGEMENT SYSTEM



AGRO-FARM

Menu
Vehicle/Equipment Type
Vehicle/Equipment Details
Employment Details
Land Details
Crop Details
Expenses
Income
Report

Vehicle / Equipment Details

Vehicle / Equipment Name :

Type Name :

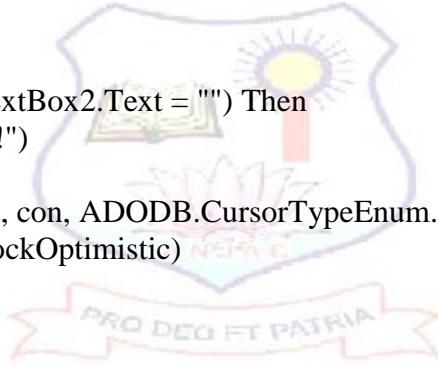
Price :

Details :

Copyright © 2021 Anjana Infotech. All rights reserved.

Vehicle / Equipment type :

If (TextBox1.Text = "" And TextBox2.Text = "") Then
 MsgBox("Enter all the details!!")
 Else
 rec.Open("select * from vtype", con, ADODB.CursorTypeEnum.adOpenDynamic,
 ADODB.LockTypeEnum.adLockOptimistic)
 rec.AddNew()
 rec(1).Value = TextBox1.Text
 rec(2).Value = TextBox2.Text
 rec.Update()
 MsgBox("Vehicle / Equipment Type Added!!!")
 rec.Close()
 End If
 Me.Close()





Menu

Vehicle/Equipment Type

Vehicle/Equipment Details

Employment Details

Land Details

Crop Details

Expenses

Income

Report

Vehicle / Equipment Type

Vehicle Type :

Details :

Copyright © 2021 Anjana Infotech. All rights reserved.



Ex. No. :11

Date : **Cargo Management System**

Aim :

To create a windows application for the maintenance of the cargo.

Program :

Login:

```
If TextBox1.Text = "admin" And TextBox2.Text = "admin" Then  
Form1.Show()  
    TextBox1.Text = ""  
    TextBox2.Text = ""  
ElseIf TextBox1.Text = "report" And TextBox2.Text = "report" Then  
Form2.Show()  
    TextBox1.Text = ""  
    TextBox2.Text = ""  
End If
```



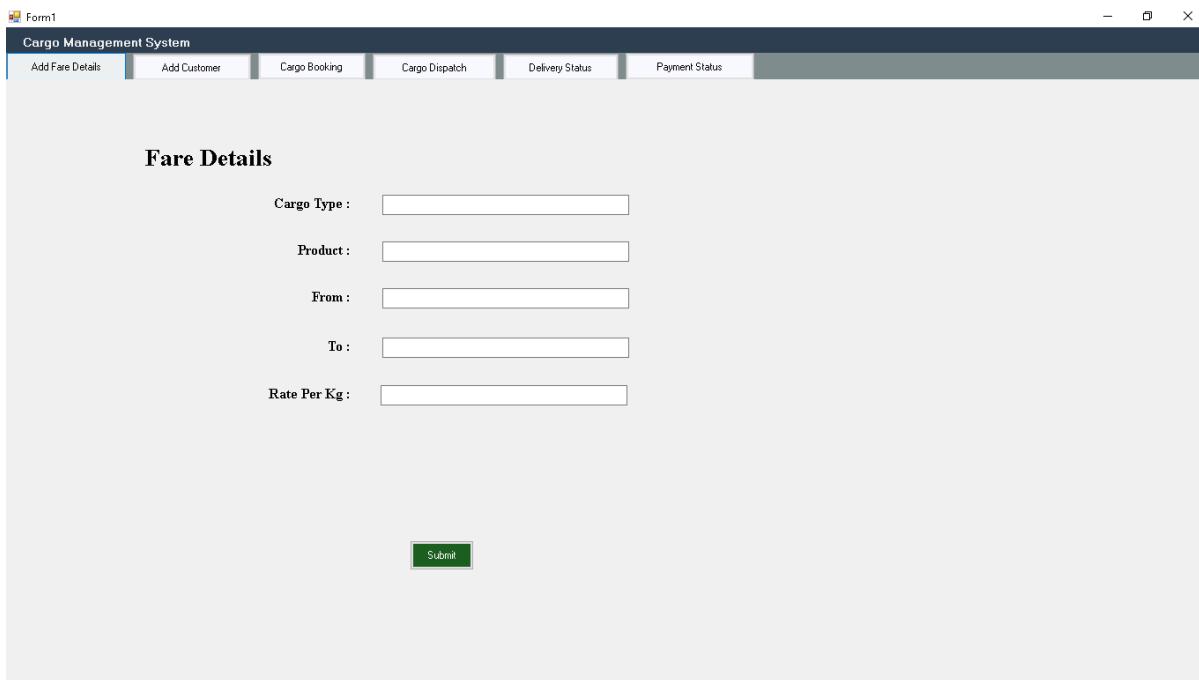
Fare Details :

```
If (TextBox3.Text = "" And TextBox2.Text = "") Then  
MsgBox("Enter all the details!!")  
Else
```

```

rec.Open("select * from fare", con, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)
rec.AddNew()
rec(1).Value = TextBox1.Text
rec(2).Value = TextBox4.Text
rec(3).Value = TextBox3.Text
rec(4).Value = TextBox2.Text
rec(5).Value = TextBox5.Text
rec.Update()
MsgBox("Cargo fare Details Added!!")
rec.Close()
End If
Me.Close()

```



Payment List:

```

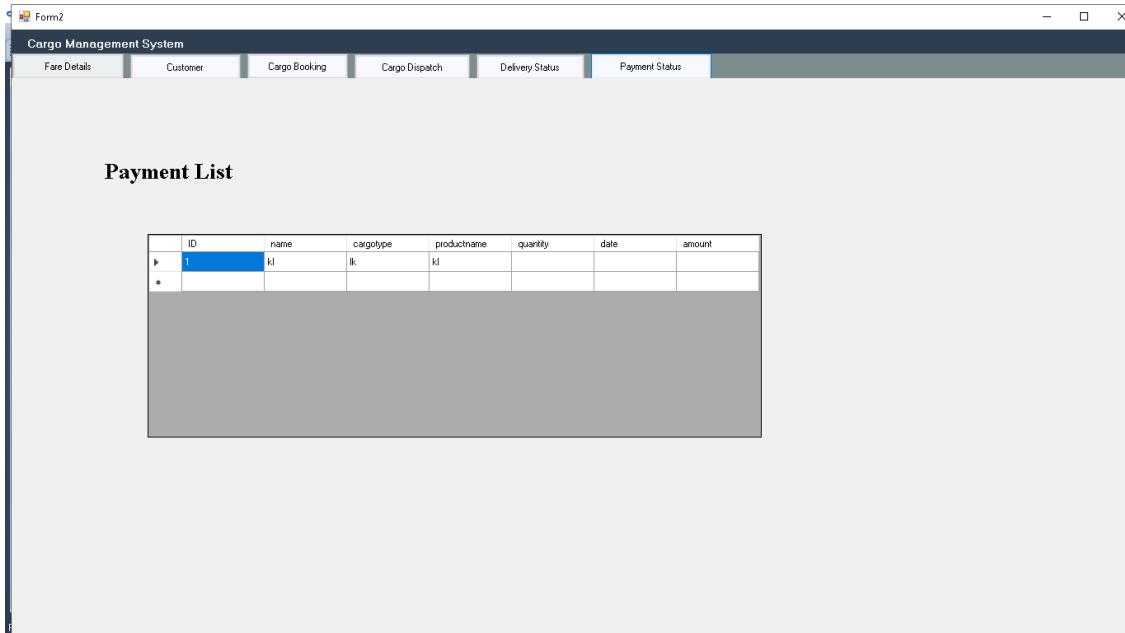
If (TextBox3.Text = "" And TextBox2.Text = "") Then
MsgBox("Enter all the details!!")
Else
rec.Open("select * from payment", con, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)
rec.AddNew()
rec(1).Value = ComboBox2.Text
rec(2).Value = ComboBox1.Text
rec(3).Value = TextBox1.Text

```

```

rec(4).Value = TextBox4.Text
rec(5).Value = TextBox3.Text
rec(6).Value = TextBox2.Text
rec.Update()
MsgBox("Cargo Payment Details Added!!")
rec.Close()
End If
Me.Close()

```



Delivery Status List:

```

If (TextBox3.Text = "" And TextBox2.Text = "") Then
    MsgBox("Enter all the details!!")
Else
    rec.Open("select * from delivery where name = " & ComboBox2.Text & "", con,
    ADODB.CursorTypeEnum.adOpenDynamic, ADODB.LockTypeEnum.adLockOptimistic)
    rec.AddNew()
    rec(1).Value = ComboBox2.Text
    rec(2).Value = Datepicker1.Text
    rec(3).Value = ComboBox1.Text
    rec(4).Value = TextBox1.Text
    rec(5).Value = ComboBox4.Text
    rec(6).Value = ComboBox5.Text
    rec(7).Value = TextBox4.Text
    rec(8).Value = TextBox3.Text
    rec(9).Value = TextBox2.Text
    rec.Update()

```

```

MsgBox("Cargo Delivery Details Added!!")
rec.Close()
End If
Me.Close()

```

ID	name	date	cargotype	productname	from	to	quantity	rate	amount
1	khhkj	Wednesday, Mar...	hk	hk	h	kh	jhkj	jhk	kj
2	kjkjk	Wednesday, Mar...	jk	jk	j	ki	lki	lkj	lkj

Cargo Dispatch List :

If (TextBox3.Text = "" And TextBox2.Text = "") Then
 MsgBox("Enter all the details!!")

```

Else
rec.Open("select * from dispatch where name = '" & ComboBox2.Text & "'", con,
ADODB.CursorTypeEnum.adOpenDynamic, ADODB.LockTypeEnum.adLockOptimistic)
rec.AddNew()
rec(1).Value = ComboBox2.Text
rec(2).Value = DateTimePicker1.Text
rec(3).Value = ComboBox1.Text
rec(4).Value = ComboBox3.Text
rec(5).Value = ComboBox4.Text
rec(6).Value = ComboBox5.Text
rec(7).Value = TextBox4.Text
rec(8).Value = TextBox3.Text
rec(9).Value = TextBox2.Text
rec.Update()
MsgBox("Cargo Delivery Details Added!!")
rec.Close()

```

```
End If  
Me.Close()
```

The screenshot shows a Windows application window titled "Form2" with the title bar "Cargo Management System". Below the title bar is a navigation menu with tabs: "Fare Details", "Customer", "Cargo Booking", "Cargo Dispatch" (which is the active tab), "Delivery Status", and "Payment Status". The main content area is titled "Cargo Dispatch List". Inside this area, there is a table with columns: ID, name, date, cargotype, productname, from, to, quantity, rate, and amount. A single row is visible in the table, with the first cell containing the value "1".

Cargo Booking List :

```
If (TextBox3.Text = "" And TextBox2.Text = "") Then  
MsgBox("Enter all the details!!")
```

```
Else  
rec.Open("select * from booking", con, ADODB.CursorTypeEnum.adOpenDynamic,  
ADODB.LockTypeEnum.adLockOptimistic)  
rec.AddNew()  
rec(1).Value = ComboBox2.Text  
rec(2).Value = DateTimePicker1.Text  
rec(3).Value = ComboBox1.Text  
rec(4).Value = TextBox1.Text  
rec(5).Value = ComboBox4.Text  
rec(6).Value = ComboBox5.Text  
rec(7).Value = TextBox4.Text  
rec(8).Value = TextBox3.Text  
rec(9).Value = TextBox2.Text  
rec.Update()  
MsgBox("Cargo Booking Details Added!!")  
rec.Close()  
End If  
Me.Close()
```

Form2

Cargo Management System

Fare Details Customer Cargo Booking Cargo Dispatch Delivery Status Payment Status

Cargo Booking List

ID	name	date	cargotype	productname	from	to	quantity	rate	amount
1	jkjhk	Wednesday, Mar...	hk	hk	jh	jkh	hkj	hkj	kj
*									

Customer List:

```
If (TextBox3.Text = "" And TextBox2.Text = "") Then
    MsgBox("Enter all the details!!")
```

```
Else
    rec.Open("select * from customer", con, ADODB.CursorTypeEnum.adOpenDynamic,
        ADODB.LockTypeEnum.adLockOptimistic)
    rec.AddNew()
    rec(1).Value = TextBox1.Text
    rec(2).Value = TextBox4.Text
    rec(3).Value = TextBox3.Text
    rec(4).Value = TextBox2.Text

    rec.Update()
    MsgBox("Cargo Customer Details Added!!")
    rec.Close()
End If
Me.Close()
```

Form2

Cargo Management System

Fare Details Customer Cargo Booking Cargo Dispatch Delivery Status Payment Status

Customer List

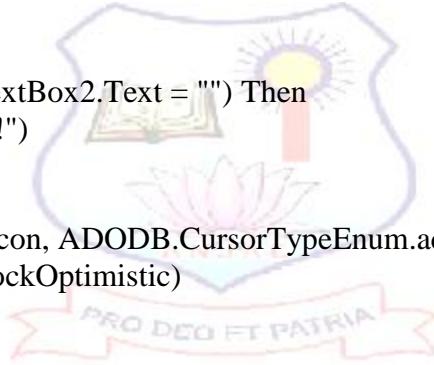
ID	name	mobile	email	address
1	jhgjh	hg	hg	gh
*				

Fare List :

```
If (TextBox3.Text = "" And TextBox2.Text = "") Then
    MsgBox("Enter all the details!!")
```

```
Else
rec.Open("select * from fare", con, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)
rec.AddNew()
rec(1).Value = TextBox1.Text
rec(2).Value = TextBox4.Text
rec(3).Value = TextBox3.Text
rec(4).Value = TextBox2.Text
rec(5).Value = TextBox5.Text
```

```
rec.Update()
MsgBox("Cargo fare Details Added!!")
rec.Close()
End If
Me.Close()
```



Form2

Cargo Management System

Fare Details Customer Cargo Booking Cargo Dispatch Delivery Status Payment Status

Fare List

ID	cargotype	product	from	to	rateperkg
1	khkj	hk	jk	hk	jk
.					

Payment List:

If (TextBox3.Text = "" And TextBox2.Text = "") Then
 MsgBox("Enter all the details!!")

Else
 rec.Open("select * from payment", con, ADODB.CursorTypeEnum.adOpenDynamic,
 ADODB.LockTypeEnum.adLockOptimistic)
 rec.AddNew()
 rec(1).Value = ComboBox2.Text
 rec(2).Value = ComboBox1.Text
 rec(3).Value = TextBox1.Text
 rec(4).Value = TextBox4.Text
 rec(5).Value = TextBox3.Text
 rec(6).Value = TextBox2.Text

rec.Update()
 MsgBox("Cargo Payment Details Added!!")
 rec.Close()
 End If
 Me.Close()

Form1

Cargo Management System

Add Fare Details Add Customer Cargo Booking Cargo Dispatch Delivery Status Payment Status

Payment Details

Name :

Cargo Type :

Product Name :

Quantity :

Received Date :

Amount :

Delivery Status :

rec.Open("select * from booking where name = " & ComboBox2.Text & "", con,
ADODB.CursorTypeEnum.adOpenDynamic, ADODB.LockTypeEnum.adLockOptimistic)

DateTimePicker1.Text = rec(2).Value

ComboBox1.Text = rec(3).Value

TextBox1.Text = rec(4).Value

ComboBox4.Text = rec(5).Value

ComboBox5.Text = rec(6).Value

TextBox4.Text = rec(7).Value

TextBox3.Text = rec(8).Value

TextBox2.Text = rec(9).Value

rec.Update()

rec.Close()

Form1

Cargo Management System

Add Fare Details Add Customer Cargo Booking Cargo Dispatch Delivery Status Payment Status

Delivery Status Details

Name :

Date :

Cargo Type :

Product Name :

From :

To :

Quantity :

Rate :

Amount :



Ex. No. :12

Date :

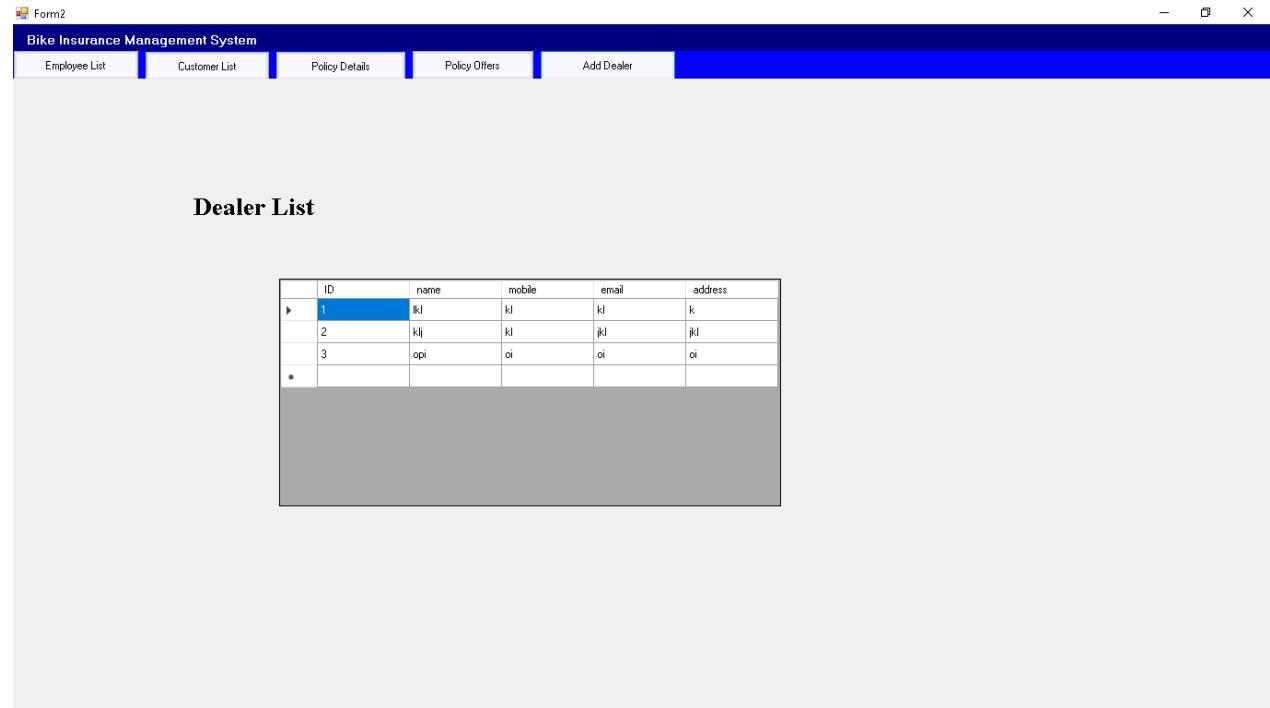
Bike Insurance Management System

Aim :

To create a windows application for the maintenance of the data in bike insurance company.

Program :

Dealers List :



Offer List :

Form2

Bike Insurance Management System

Employee List Customer List Policy Details Policy Offers Add Dealer

Offer List

ID	type	offer
1	oiu	ui
*		

Policy List :

Form2

Bike Insurance Management System

Employee List Customer List Policy Details Policy Offers Add Dealer

Policy List

ID	clientno	clientname	policytype	paidamount	count	policyamount	lastpayment	balance
1	itui	u	ui	iu	i	ui	ui	ui
*								

Customer List :

Form2

Bike Insurance Management System

Employee List Customer List Policy Details Policy Offers Add Dealer

Customer List

ID	name	mobile	email	dob	address
1	jh	jh	i	Thursday, March ... hi	
2	kjh	jk	hi	Thursday, March ... hkj	
3	oiuiu	iou	io	Friday, March 26... uo	
*					

Employee List :

Form2

Bike Insurance Management System

Employee List Customer List Policy Details Policy Offers Add Dealer

Employee List

ID	name	userrole	mobile	email	dob	address
1	ggh	h	hg	hg	Thursday, March ... hg	
2	kij	jk	ki	kj	Thursday, March ... k	
3	kij	jk	ki	k	Friday, March 26... jk	
*						

Add Dealers :

```
rec.Open("select * from dealer", con, ADODB.CursorTypeEnum.adOpenDynamic,  
ADODB.LockTypeEnum.adLockOptimistic)  
rec.AddNew()  
rec(1).Value = TextBox1.Text  
rec(2).Value = TextBox3.Text  
rec(3).Value = TextBox4.Text  
rec(4).Value = TextBox5.Text  
rec.Update()  
MsgBox("Bike Dealer Details Added!!")  
rec.Close()  
Me.Close()
```



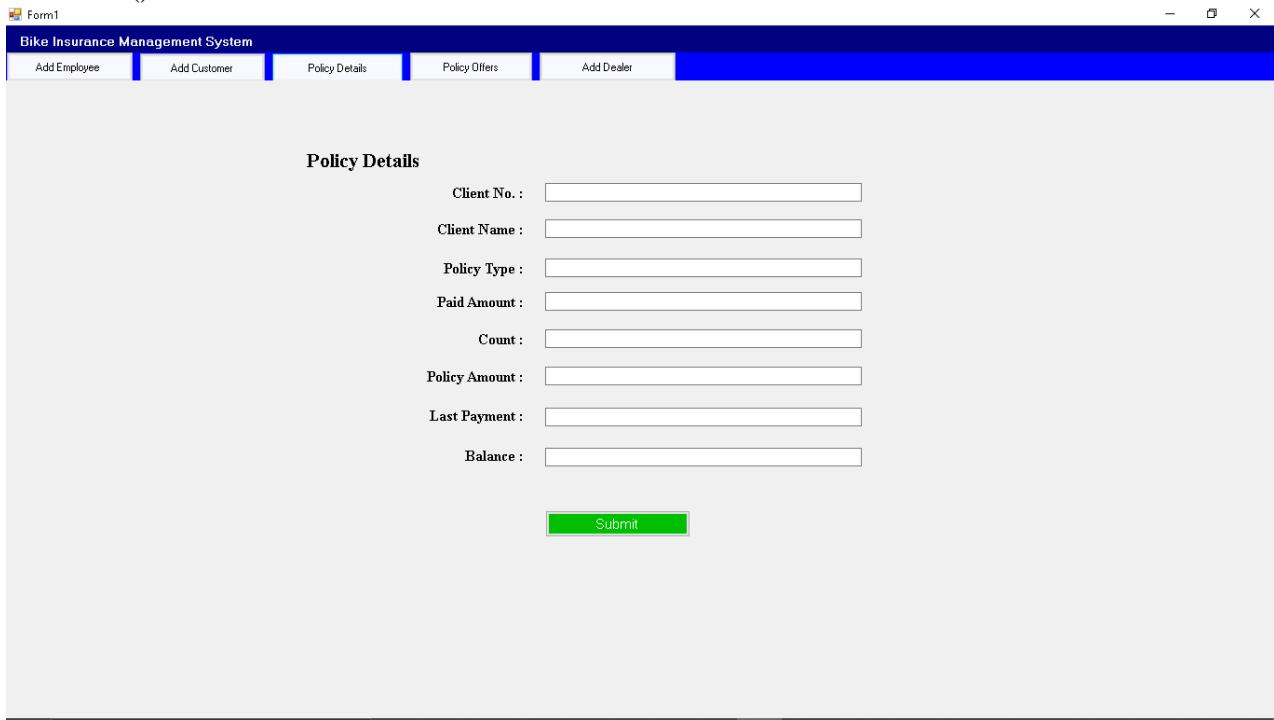
Policy Details :

```
rec.Open("select * from policy", con, ADODB.CursorTypeEnum.adOpenDynamic,  
ADODB.LockTypeEnum.adLockOptimistic)  
rec.AddNew()  
rec(1).Value = TextBox1.Text  
rec(2).Value = TextBox3.Text  
rec(3).Value = TextBox8.Text  
rec(4).Value = TextBox4.Text  
rec(5).Value = TextBox5.Text  
rec(6).Value = TextBox7.Text  
rec(7).Value = TextBox6.Text
```

```

rec(8).Value = TextBox2.Text
rec.Update()
MsgBox("Policy Details Added!!")
rec.Close()
Me.Close()

```



Add Customer :

```

rec.Open("select * from customer", con, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)
rec.AddNew()
rec(1).Value = TextBox1.Text
rec(2).Value = TextBox3.Text
rec(3).Value = TextBox4.Text
rec(4).Value = DateTimePicker1.Text
rec(5).Value = TextBox5.Text
rec.Update()
MsgBox("Customer Details Added!!")
rec.Close()
Me.Close()

```

Form1

Bike Insurance Management System

- Add Employee
- Add Customer
- Policy Details
- Policy Offers
- Add Dealer

Add Customer

Name :

Mobile :

E-Mail :

D.O.B. :

Address :

Add Employee :

```
rec.Open("select * from employee", con, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)
rec.AddNew()
```

```
rec(1).Value = TextBox1.Text
rec(2).Value = ComboBox1.Text
rec(3).Value = TextBox3.Text
rec(4).Value = TextBox4.Text
rec(5).Value = DateTimePicker1.Text
rec(6).Value = TextBox5.Text
rec.Update()
MsgBox("Employee Details Added!!")
rec.Close()
Me.Close()
```

Form1

Bike Insurance Management System

Add Employee Add Customer Policy Details Policy Offers Add Dealer

Add Employee

Name :

User Role :

Mobile :

E-Mail :

D.O.B : 10 November 2021

Address :

Login :

If TextBox1.Text = "admin" And TextBox2.Text = "admin" Then
Form1.Show()
 TextBox1.Text = ""
 TextBox2.Text = ""
ElseIf TextBox1.Text = "report" And TextBox2.Text = "report" Then
Form2.Show()
 TextBox1.Text = ""
 TextBox2.Text = ""
End If



login

- □ X

LOGIN

User ID :

Password :



Ex. No. :13

Date :

Voting Management System

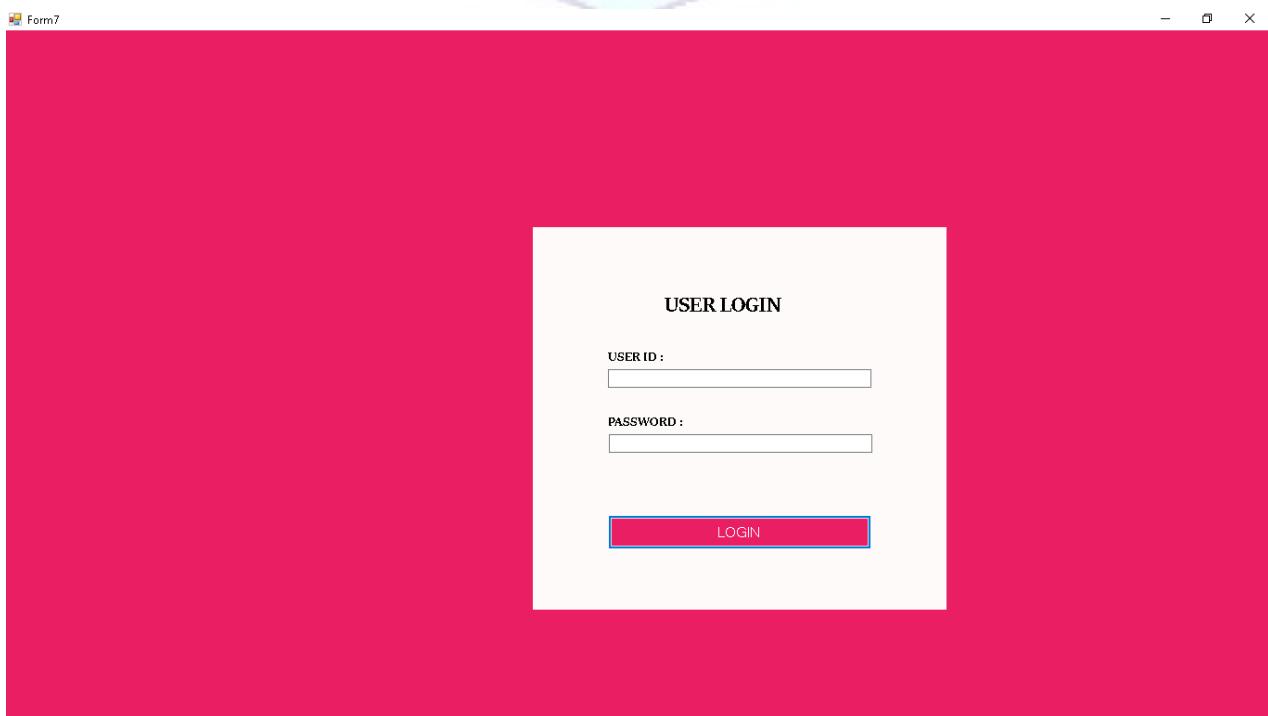
Aim :

To create a windows application for the voting system.

Program :

Login :

```
If TextBox1.Text = "Admin" And TextBox2.Text = "Admin" Then  
Form5.Show()  
Me.Hide()  
    TextBox1.Text = ""  
    TextBox2.Text = ""  
ElseIf TextBox1.Text = "User" And TextBox2.Text = "User" Then  
Form1.Show()  
Me.Hide()  
    TextBox1.Text = ""  
    TextBox2.Text = ""  
Else  
MsgBox("Login Failed")  
End If
```



OTP Verification :

Try

```
Dim Smtp_ServerAs New SmtpClient  
Dim e_mail As New MailMessage  
Smtp_Server.UseDefaultCredentials = False  
Smtp_Server.Credentials = New Net.NetworkCredential("banuganapathy99@gmail.com",  
"santhi99")  
Smtp_Server.Port = 587  
Smtp_Server.EnableSsl = True  
Smtp_Server.Host = "smtp.gmail.com"  
e_mail = New MailMessage()  
e_mail.From = New MailAddress(TextBox3.Text)  
e_mail.To.Add(TextBox1.Text)  
e_mail.Subject = TextBox2.Text  
e_mail.IsBodyHtml = False  
e_mail.Body = "Your OTP for Online Voting is " + TextBox4.Text  
Smtp_Server.Send(e_mail)  
MsgBox("Send")  
Catch ex As Exception  
MsgBox(ex.Message)
```

End Try



Enter Verification Code :

Generate OTP

Send OTP

OK Cancel

Upload Details :

```
r.Open("select * from Candidate", c, ADODB.CursorTypeEnum.adOpenDynamic,  
ADODB.LockTypeEnum.adLockOptimistic)  
r.AddNew()  
r(1).Value = TextBox1.Text  
r(2).Value = TextBox2.Text  
r(3).Value = TextBox3.Text  
r(4).Value = TextBox4.Text  
r(5).Value = TextBox5.Text  
r(6).Value = TextBox6.Text  
r(7).Value = TextBox7.Text  
r.Update()  
r.Close()  
MsgBox("Confirm Your OTP!!")  
Me.Hide()  
Form5.Show()
```



Upload AadharID :

```
Me.OpenFileDialog1.ShowDialog()  
Me.PictureBox1.Load(Me.OpenFileDialog1.FileName)  
TextBox1.Text = OpenFileDialog1.FileName
```



Upload Voter ID :

```
Me.OpenFileDialog1.ShowDialog()  
Me.PictureBox1.Load(Me.OpenFileDialog1.FileName)  
TextBox1.Text = OpenFileDialog1.FileName
```

ONLINE VOTING SYSTEM

Upload Voter Id :



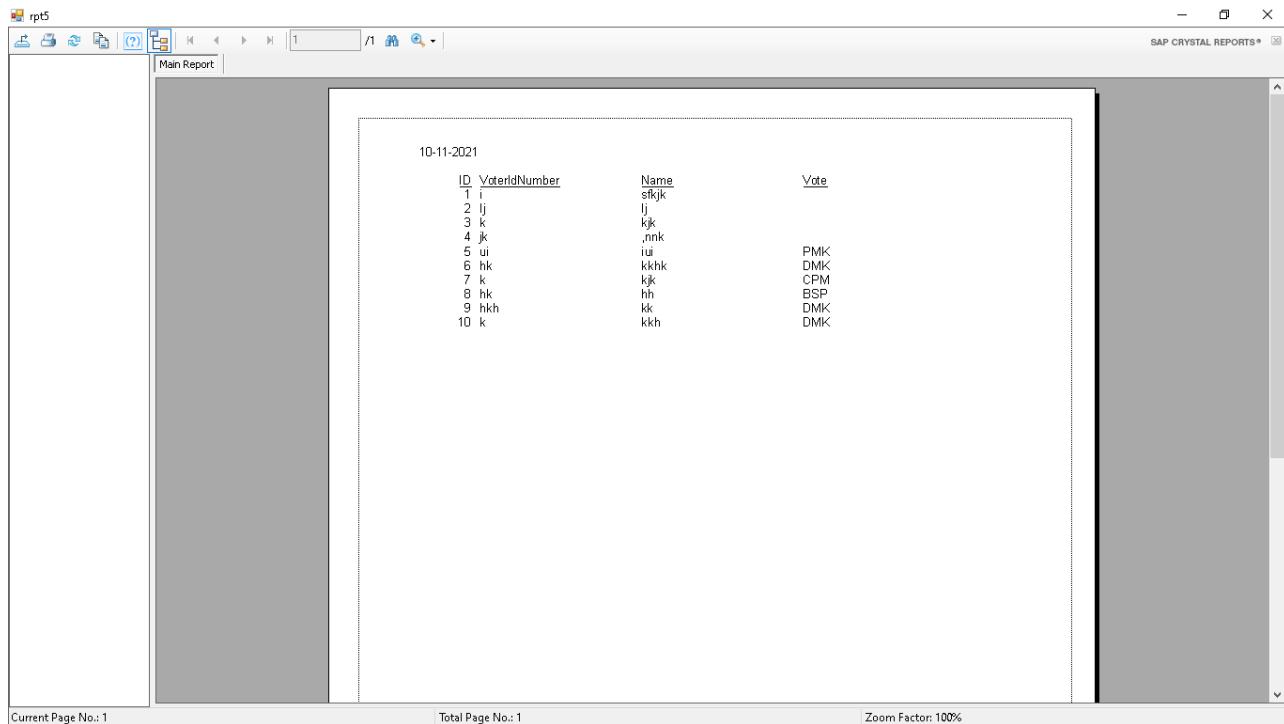
Location Wise Report :

SAP CRYSTAL REPORTS®

Current Page No.: 1 Total Page No.: 1 Zoom Factor: 100%

ID	Address	Name	Vote
1	ij	stfjk	
2	l	lj	
3	jk	kjk	
4	jk	,nnk	
5	u	ui	PMK
6	h	kkhk	DMK
7	kj	kjk	CPM
8	h	hh	BSP
9	k	kk	DMK
10	ih	kkh	DMK

Voter ID Report :



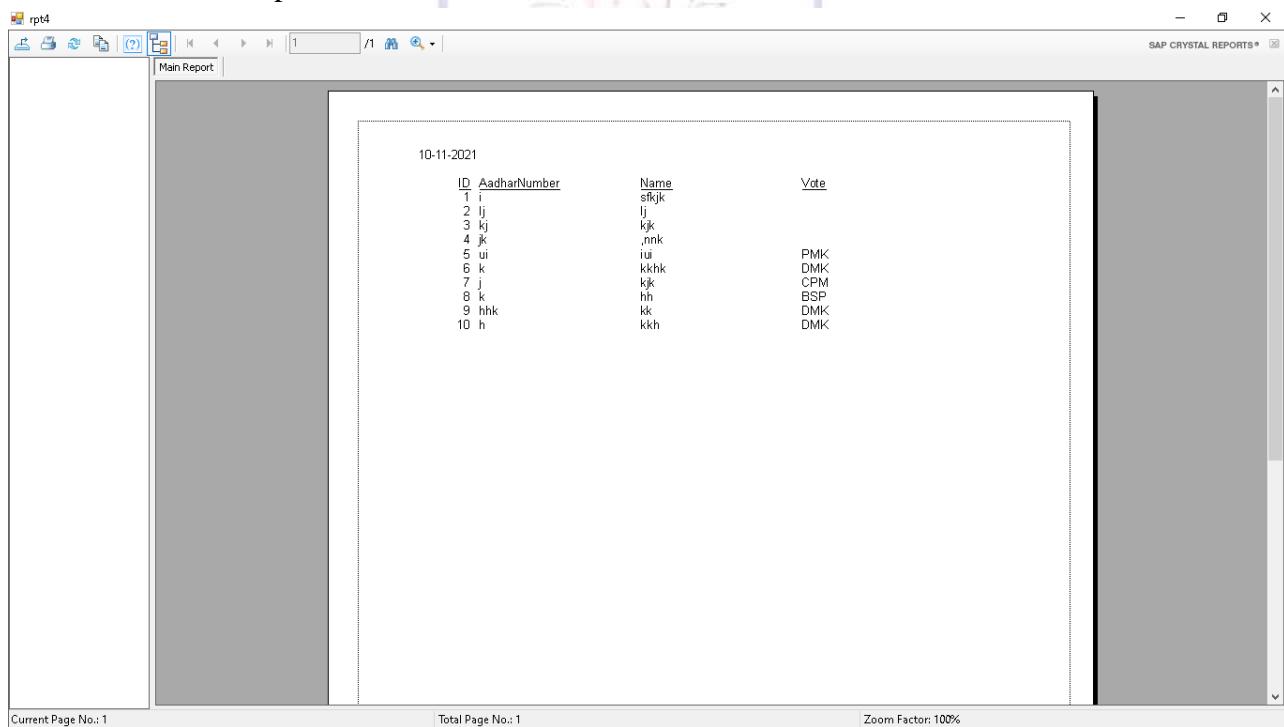
The screenshot shows the SAP Crystal Reports software interface for a report titled "Voter ID Report". The report is displayed in a main window with a light gray background. At the top left, there is a toolbar with various icons for file operations like Open, Save, Print, and Zoom. The title bar "Main Report" is visible. In the center, a table is presented with the following data:

ID	VoterIdNumber	Name	Vote
1	i	sfkjk	
2	lj	lj	
3	k	kjk	
4	jk	,nnk	
5	ui	iui	PMK
6	hk	kkhk	DMK
7	k	kjk	CPM
8	hk	hh	BSP
9	hh	kk	DMK
10	k	kkh	DMK

At the bottom of the report area, there is a decorative watermark or logo.

Current Page No.: 1 Total Page No.: 1 Zoom Factor: 100%

Aadhar No. Wise Report :



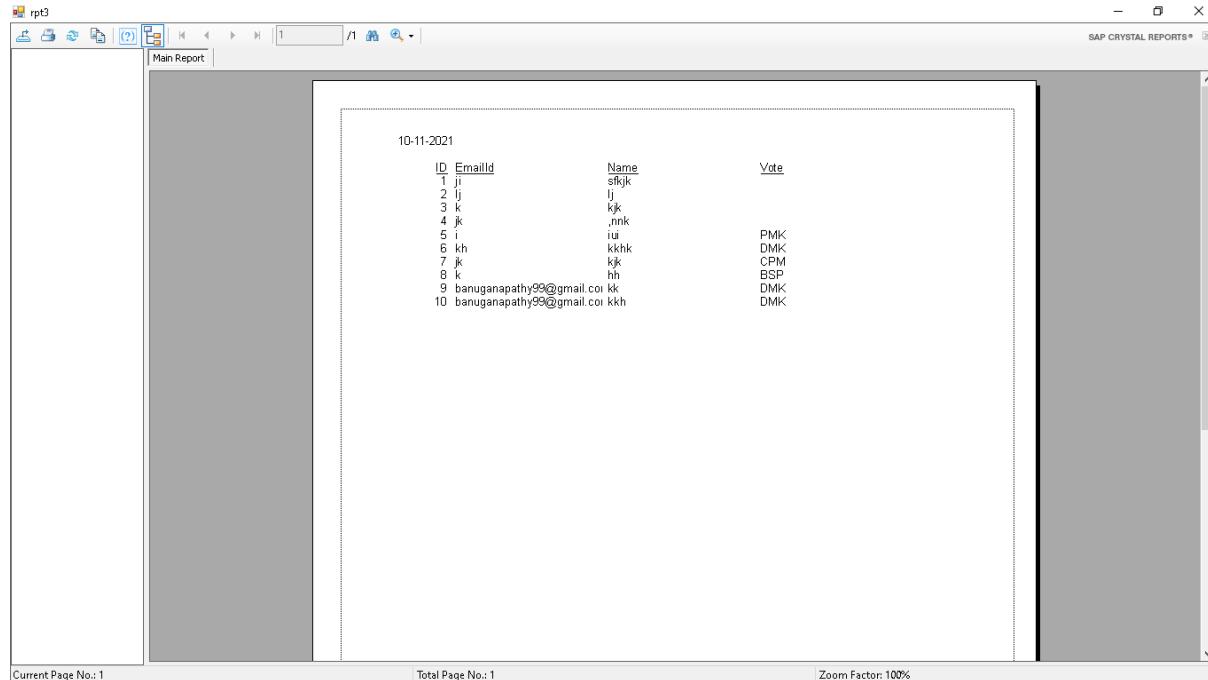
The screenshot shows the SAP Crystal Reports software interface for a report titled "Aadhar No. Wise Report". The report is displayed in a main window with a light gray background. At the top left, there is a toolbar with various icons for file operations like Open, Save, Print, and Zoom. The title bar "Main Report" is visible. In the center, a table is presented with the following data:

ID	AadharNumber	Name	Vote
1	i	sfkjk	
2	lj	lj	
3	kj	kjk	
4	jk	,nnk	
5	ui	iui	PMK
6	k	kkhk	DMK
7	j	kjk	CPM
8	k	hh	BSP
9	hh	kk	DMK
10	h	kkh	DMK

At the bottom of the report area, there is a decorative watermark or logo.

Current Page No.: 1 Total Page No.: 1 Zoom Factor: 100%

Email ID Wise Report :

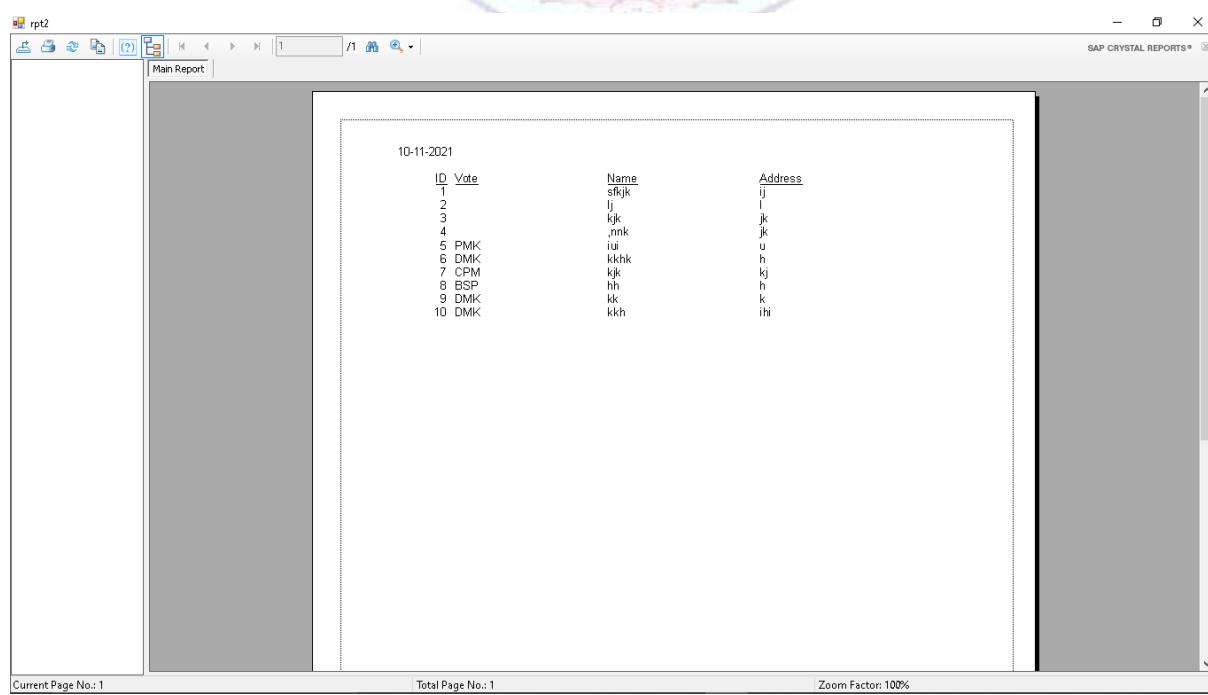


The screenshot shows the SAP Crystal Reports software interface. The title bar says "rpt3" and "SAP CRYSTAL REPORTS". The main window is titled "Main Report". Inside the report area, there is a table with the following data:

ID	EmailId	Name	Vote
1	ji	sjk	
2	ij	ij	
3	k	kk	
4	jk	,nnk	
5	i	iu	PMK
6	kh	kkhk	DMK
7	jk	jk	CPM
8	k	hh	BSP
9	banuganapathy99@gmail.com	kk	DMK
10	banuganapathy99@gmail.com	kkh	DMK

At the bottom of the report, there are status bars: "Current Page No.: 1", "Total Page No.: 1", and "Zoom Factor: 100%".

Voter Id Wise Report :



The screenshot shows the SAP Crystal Reports software interface. The title bar says "rpt2" and "SAP CRYSTAL REPORTS". The main window is titled "Main Report". Inside the report area, there is a table with the following data:

ID	Vote	Name	Address
1	sjk	ij	
2		l	
3		jk	
4		,nnk	
5	PMK	iu	u
6	DMK	kkhk	h
7	CPM	jk	kj
8	BSP	hh	h
9	DMK	kk	k
10	DMK	kkh	ih

At the bottom of the report, there are status bars: "Current Page No.: 1", "Total Page No.: 1", and "Zoom Factor: 100%".

Name Wise Report :

SAP CRYSTAL REPORTS®

Main Report

10-11-2021

ID	Name	Dob	EmailId	AadharNumber	VoterIdNumber	Address	Vote
1	sjkjk	hk	ji	i	ij	ij	
2	jj	lj	lj	lj	lj	lj	
3	jkj	j	k	jk	k	jk	
4	jkjk	k	jk	jk	jk	jk	
5	ju	lu	ui	ui	u	u	PMK
6	hkhk	h	kh	k	hk	h	DMK
7	hjk	jk	jk	j	kj	kj	CPM
8	hh	hkhk	k	k	hk	h	BSP
9	kk	k	banuganapathy	hhk	hhk	k	DMK
10	kkh	kh	banuganapathy	h	k	hi	DMK

Current Page No.: 1

Total Page No.: 1

Zoom Factor: 100%



Ex. No. :14

Date :

Software Learning System

Aim :

To create a windows application for the software learning system.

Program :

Contact Details :



Co Curricular Report :

Form1

GET ME A COURSE
Learning Made Easy!

Menu

- Home
- Add Student
- Add Course
- View Course
- Logout

Report

- Student Report
- Staff Report
- Academic Report
- Personal Details
- Studies Report
- Co Curricular Details

Co Curricular Report

ID	name	dateofadmission	class
1	ret	tret	ert
2	tt	tr	ert
3	kjhjh	Friday, March 26... 2021	hj
*			

Studies Report :

Form1

GET ME A COURSE
Learning Made Easy!

Menu

- Home
- Add Student
- Add Course
- View Course
- Logout

Report

- Student Report
- Staff Report
- Academic Report
- Personal Details
- Studies Report
- Co Curricular Details

Studies Report

ID	name	dateofadmission	cocurricular	institutename	class
1	ret	tret	terre	ert	ert
2	tt	tr	etr	ert	ert
3	kjhjh	Friday, March 26... 2021	hj	i	hj
*					

Personal Details :

Form1

GET ME A COURSE
Learning Made Easy !

Menu

- Home
- Add Student
- Add Course
- View Course
- Logout

Report

- Student Report
- Staff Report
- Academic Report
- Personal Details
- Studies Report
- Co Curricular Details
- Contact Details

Personal Details

ID	name	cocurricular	address	dob	gender	mobileno
1	ret	telre	tretr	elretr	etr	ert
2	tt	etr	ett	etr	e	rt
3	kjhjh	hj	h	Friday, March 26, 2021	jh	jh
*						

Academic Reports :

Form1

GET ME A COURSE
Learning Made Easy !

Menu

- Home
- Add Student
- Add Course
- View Course
- Logout

Report

- Student Report
- Staff Report
- Academic Report
- Personal Details
- Studies Report
- Co Curricular Details
- Contact Details

Academic Report

ID	name	institutename	class	fees
1	ret	etr	ert	ert
2	tt	etr	ert	ert
3	kjhjh	j	hj	j
*				

Staff Reports :

Form1

GET ME A COURSE
Learning Made Easy !

Menu

- Home
- Add Student
- Add Course
- View Course
- Logout

Report

- Student Report
- Staff Report**
- Academic Report
- Personal Details
- Studies Report
- Co Curricular Details
- Contact Details

Staff Report

ID	name	class	date	qualification	salary
1	jk	lk	Friday, March 26...	jk	jk
*					

Student Report :

Form1

GET ME A COURSE
Learning Made Easy !

Menu

- Home
- Add Student
- Add Course
- View Course
- Logout

Report

- Student Report**
- Staff Report
- Academic Report
- Personal Details
- Studies Report
- Co Curricular Details
- Contact Details

Students Report

ID	name	dateofadmission	cocurricular	address	dob	gender
1	ret	lret	tete	tretr	etetr	etr
2	tt	tr	elt	eit	ert	e
3	kjhjh	Friday, March 26...	hi	h	Friday, March 26...	jh
*						

View Content :

```
If (ComboBox1.Text = "MS Excel" And ComboBox2.Text = "Create Worksheet") Then  
o1.Show()  
o7.Show()  
o8.Show()  
ElseIf (ComboBox1.Text = "MS Excel" And ComboBox2.Text = "Copy Worksheet") Then  
o2.Show()  
o9.Show()  
o10.Show()  
o11.Show()  
ElseIf ComboBox1.Text = "MS Excel" And ComboBox2.Text = "Hide Worksheet" Then  
o3.Show()  
o12.Show()  
ElseIf ComboBox1.Text = "MS Excel" And ComboBox2.Text = "Delete Worksheet" Then  
o4.Show()  
o13.Show()  
ElseIf ComboBox1.Text = "MS Excel" And ComboBox2.Text = "Close Worksheet" Then  
o5.Show()  
o14.Show()  
ElseIf ComboBox1.Text = "MS Excel" And ComboBox2.Text = "Save Worksheet" Then  
o6.Show()  
o15.Show()  
End If
```



Form1

Add Course :

```
If (TextBox1.Text = "" And ComboBox3.Text = "") Then  
MsgBox("Enter all the details!!")
```

```
Else
```

```
rec.Open("select * from staff", con, ADODB.CursorTypeEnum.adOpenDynamic,  
ADODB.LockTypeEnum.adLockOptimistic)
```

```
rec.AddNew()
```

```
rec(1).Value = TextBox1.Text
```

```
rec(2).Value = ComboBox3.Text
```

```
rec(3).Value = Datepicker1.Text
```

```
rec(4).Value = ComboBox1.Text
```

```
rec(5).Value = ComboBox2.Text
```

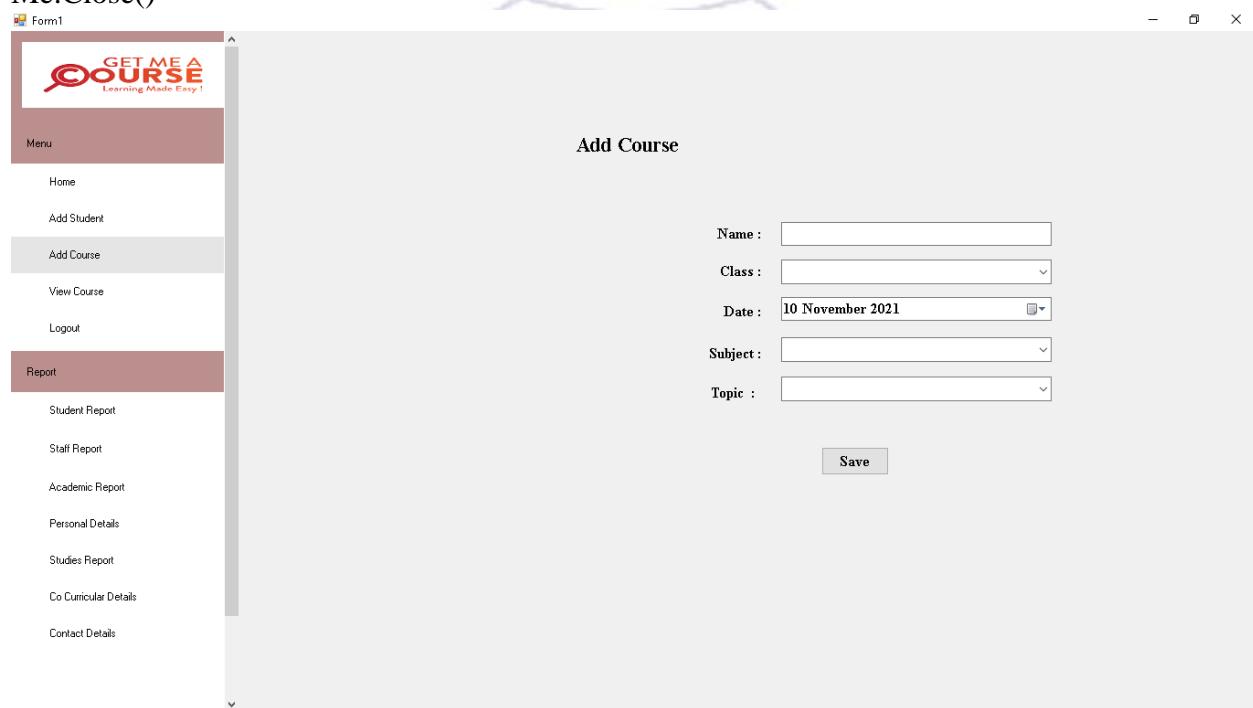
```
rec.Update()
```

```
MsgBox("Staff Added!!")
```

```
rec.Close()
```

```
End If
```

```
Me.Close()
```



Add Admission :

```
If (TextBox1.Text = "" And TextBox3.Text = "" And ComboBox4.Text = "") Then  
MsgBox("Enter all the details!!")
```

```
Else
```

```

rec.Open("select * from admission", con, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)

rec.AddNew()
rec(1).Value = TextBox1.Text
rec(2).Value = DateTimePicker1.Text
rec(3).Value = ComboBox3.Text
rec(4).Value = TextBox2.Text
rec(5).Value = DateTimePicker2.Text
rec(6).Value = ComboBox4.Text
rec(7).Value = TextBox3.Text
rec(8).Value = TextBox5.Text
rec.Update()
MsgBox("New student Added!!")
rec.Close()
Me.Close()
End If

```

Me.Close()

Front Page :



ASP.NET

ASP.NET provides basic and advanced concepts of ASP.NET. Our ASP.NET Tutorial is designed for beginners and professionals both.

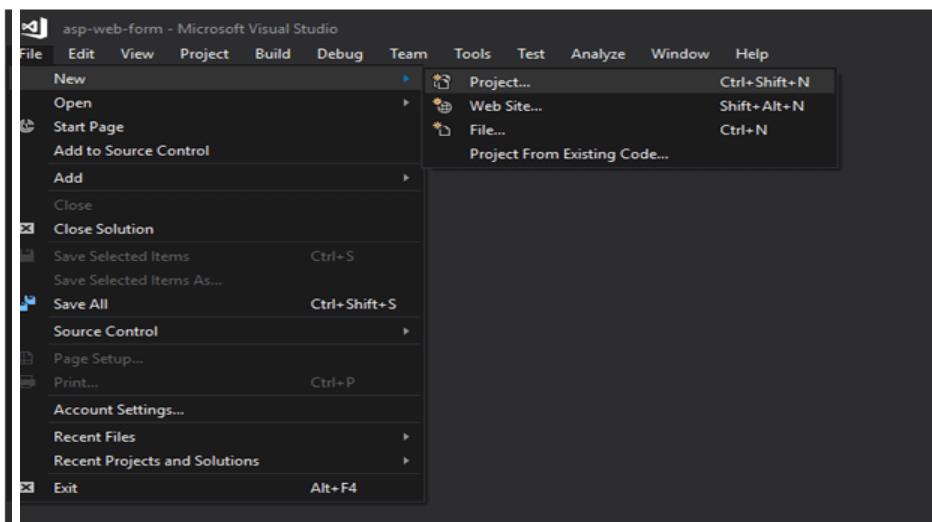
ASP.NET is a web framework designed and developed by Microsoft. It is used to develop websites, web applications and web services. It provides fantastic integration of HTML, CSS and JavaScript. It was first released in January 2002. It is built on the Common Language Runtime (CLR) and allows programmers to write code using any supported .NET language.

Our ASP.NET includes all topics of ASP.NET Tutorial such as ASP.Net introduction, features, project, example, server controls, labels, textbox, button, hyperlink, radiobutton, calender, checkbox, fileupload, events handling, authentication, webforms model binding, html server control, compare validdator, range validator, validation summary, mvc introduction, mvc project, view, validation, entity framework, authentication etc..

ASP.NET Web Forms Project

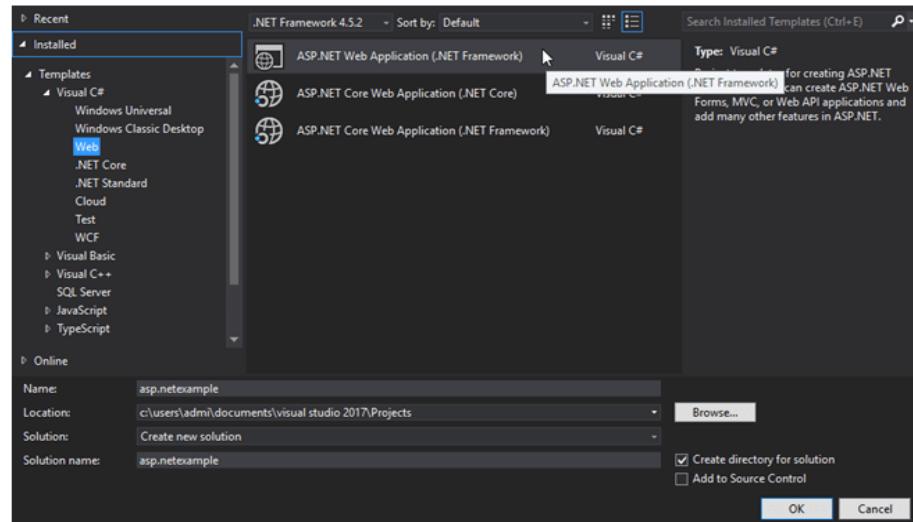
We are using Visual studio 2017 to create web project. It includes the following steps:

1. Creating a new project. Click on the file menu from the menu bar and select **new -> project.**



Select Project type

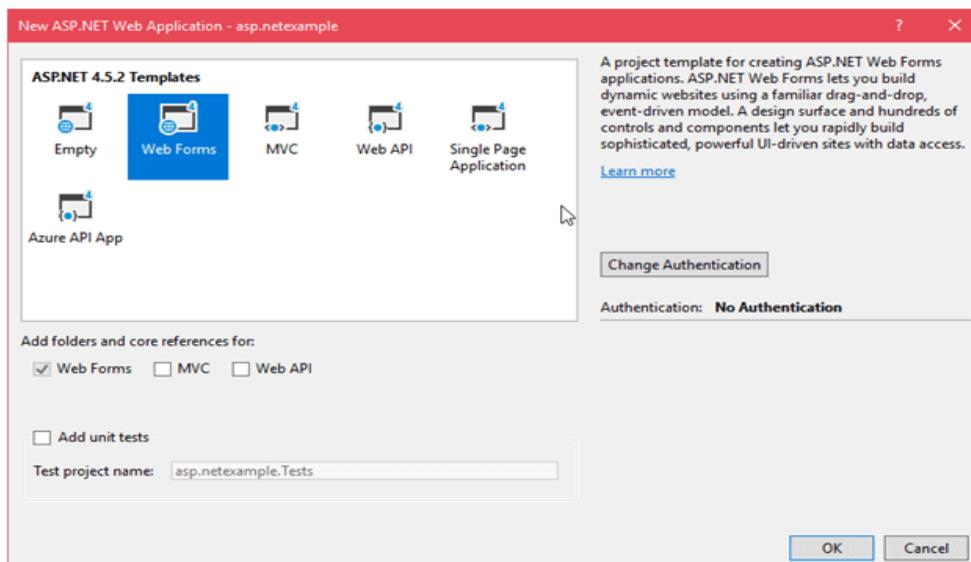
It provides couple of choices but we selecting ASP.NET Web Application.



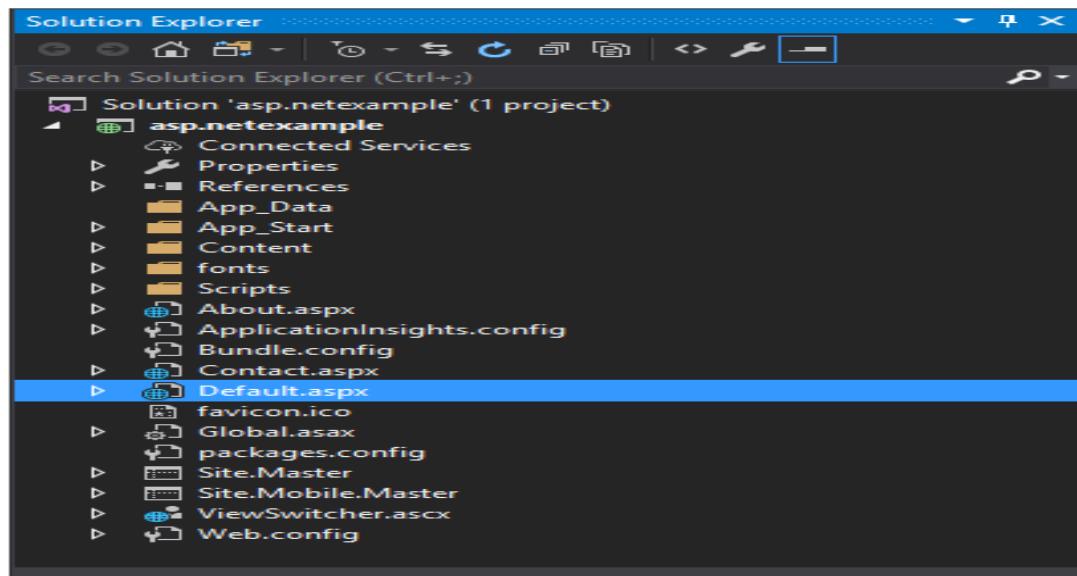
Select Project Template

After selecting project types, now, it asks for the type of template that we want to implement in our application.

Here, we are selecting Web Forms as because we are creating a Web Forms application.

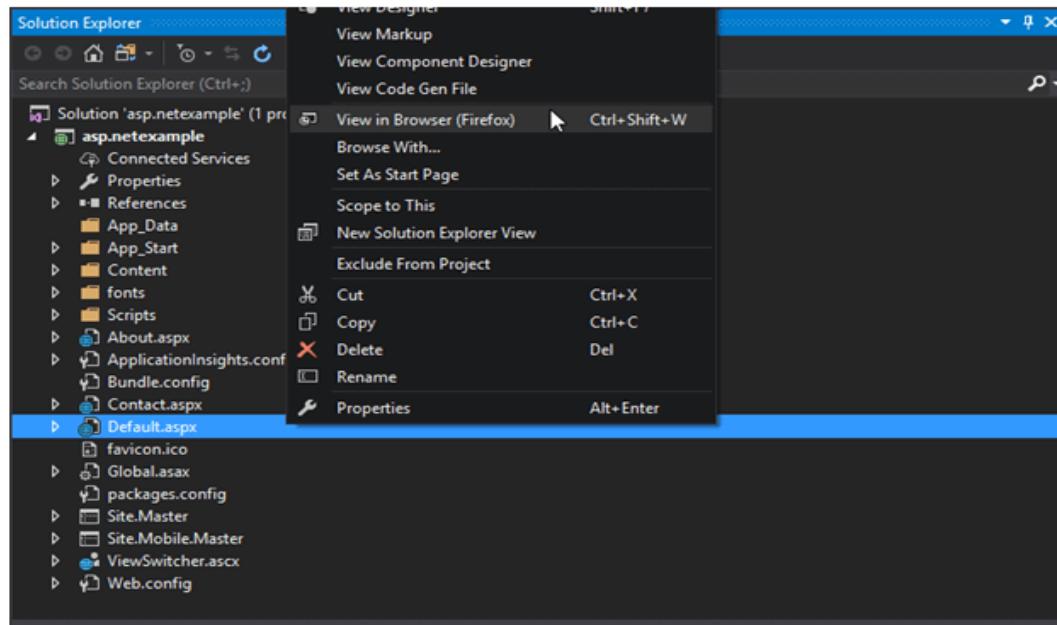


After clicking ok, it shows project in **solution explorer** window that looks like the below.

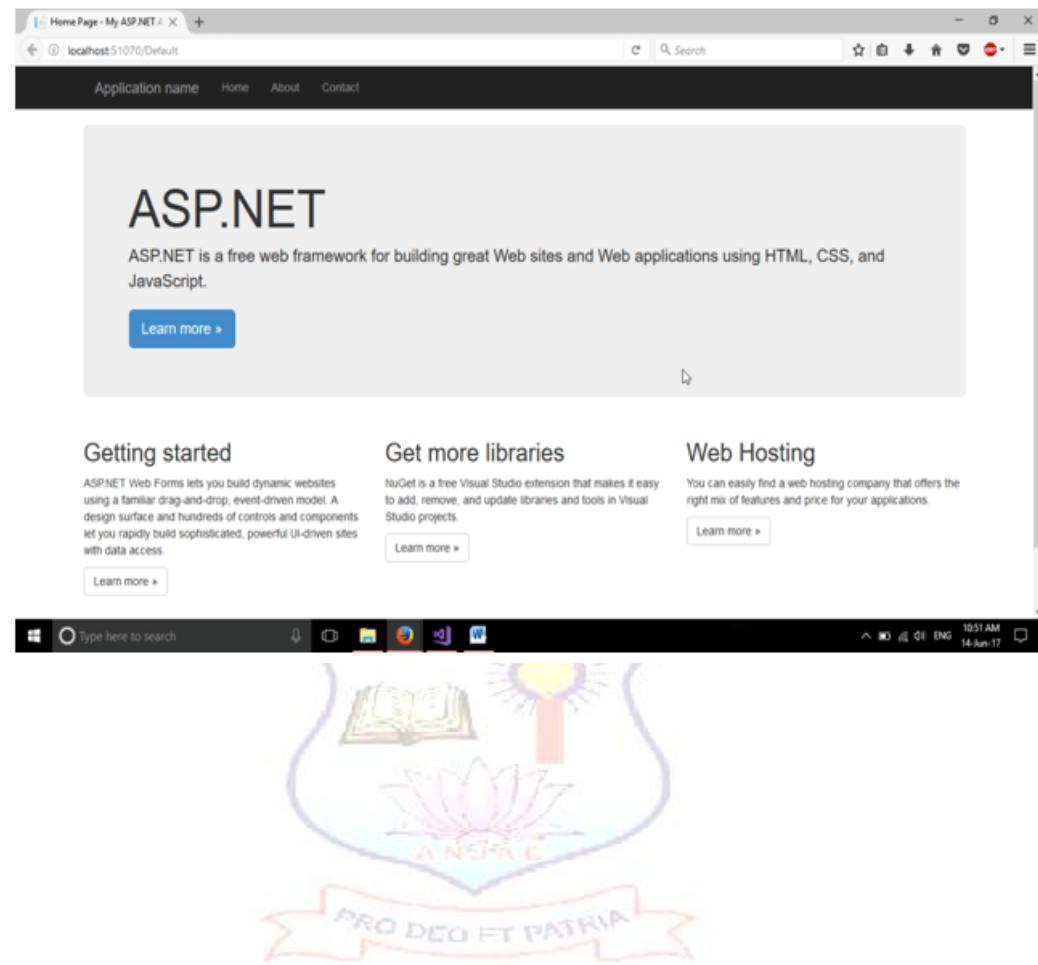


This project contains a **default.aspx** file which is a startup file. When we run the project this file executes first and display a home page of the site.

We can see its output on the browser by selecting **view in browser** option as we did below.



Finally, it shows output in the browser like this:



EX.NO:15

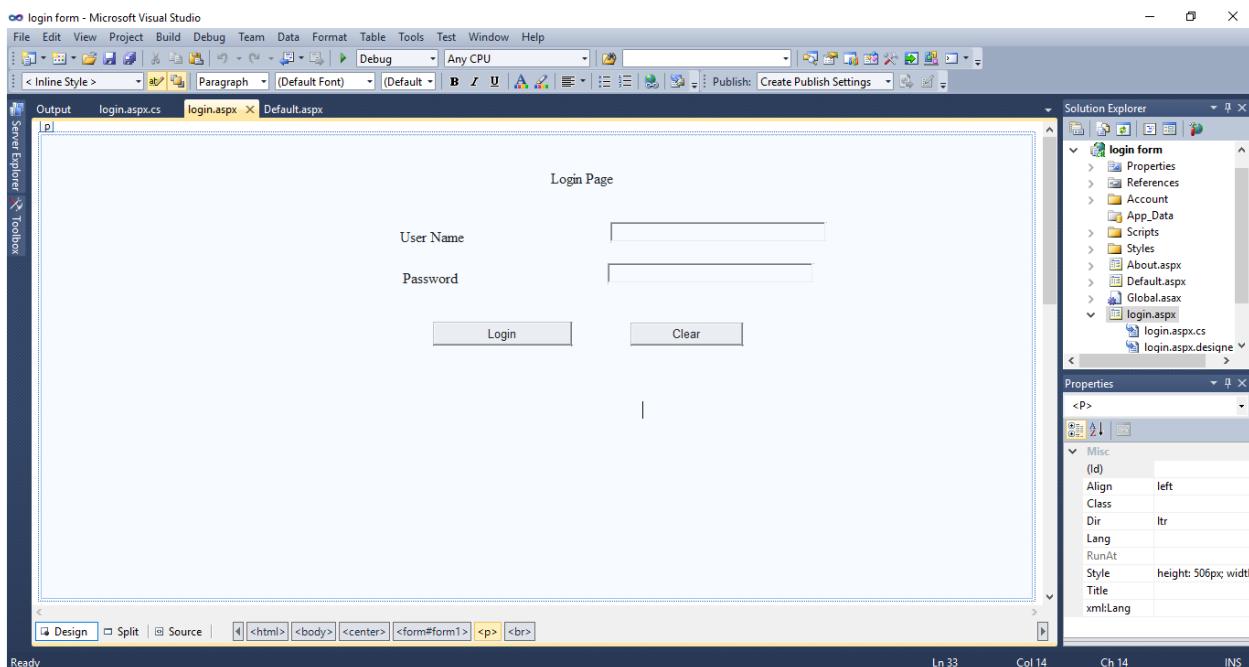
Date:

LOGIN FORM

Aim

To write a Web Application program to perform login page in asp.net

Design Window



Source Coding

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="login.aspx.cs"
Inherits="login_form.login" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    </head>
<body>
<center>
    <form id="form1" runat="server">
```


Program Coding

```
using System;
using System.Collections.Generic;
using System.Linq;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Data.OleDb;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

```
namespace login_form  
{
```

```

public partial class login : System.Web.UI.Page
{
    OleDbConnection con = new OleDbConnection();
    protected void Page_Load(object sender, EventArgs e)
    {

    }

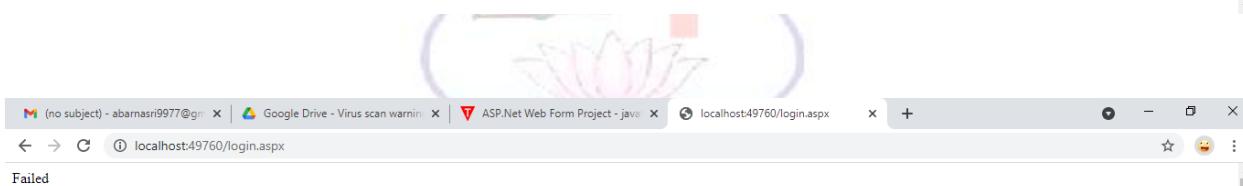
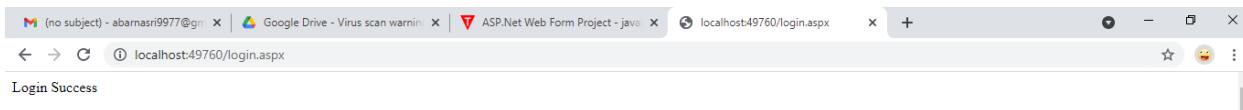
    protected void Button1_Click(object sender, EventArgs e)
    {
        OleDbCommand cmd = new OleDbCommand();
        cmd.Connection = con;
        cmd.Parameters.AddWithValue("a1", TextBox1.Text);
        cmd.Parameters.AddWithValue("a2", TextBox2.Text);
        if (TextBox1.Text == "Admin" && TextBox2.Text == "Admin123")
        {
            Response.Write("Login Success");
        }
        else
        {
            Response.Write("Failed");
        }
        con.Close();
    }

    protected void Button2_Click(object sender, System.EventArgs e)
    {
        TextBox1.Text = "";
        TextBox2.Text = "";
    }
}

```

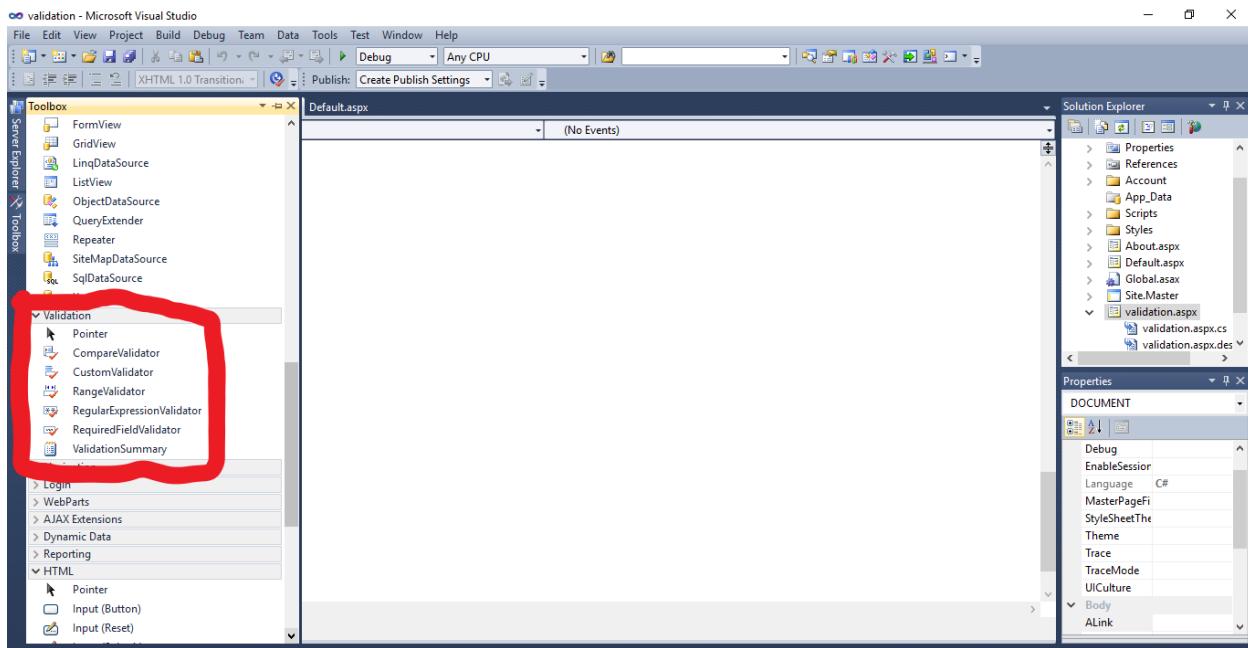


Output Window



Validation Control

1. Click view-> toolbox->validation controls



Validation controls :

1. RequiredFieldValidator:

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
    ErrorMessage="Please enter the name" ControlToValidate="TextBox1"
    ForeColor="Red" SetFocusOnError="true" Display="Dynamic"></asp:RequiredFieldValidator>
    <br />
```

2. RangeFieldValidator:

```
<asp:RangeValidator ID="RangeValidator1" runat="server" MinimumValue="20"
    MaximumValue="40"
    ErrorMessage="Please enter age only 20 to 40" ControlToValidate="TextBox2"
    ForeColor="Red" SetFocusOnError="true" Display="Dynamic"></asp:RangeValidator>
    <br />
```

3. RegularExpressionValidator:

```
<asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server"
    ErrorMessage="please enter a valid mail ID" ControlToValidate="TextBox4"
    validationExpression= "\w+([-.\w+]*)*\@(\w+([-.\w+]*)*\.\w+([-.\w+]*)*"
    ForeColor="Red" SetFocusOnError="true" Display="Dynamic">
    </asp:RegularExpressionValidator> <br />
```

4.Compare Validator:

```
<asp:TextBox ID="TextBox5" runat="server" Width="262px"></asp:TextBox>
<asp:CompareValidator ID="CompareValidator1" runat="server"
    ErrorMessage="Password and conform password does not match" ForeColor="Red"
    ControlToCompare="TextBox6" ControlToValidate="TextBox5"
    Display="Dynamic"></asp:CompareValidator>
<br />
```



EX.NO:16

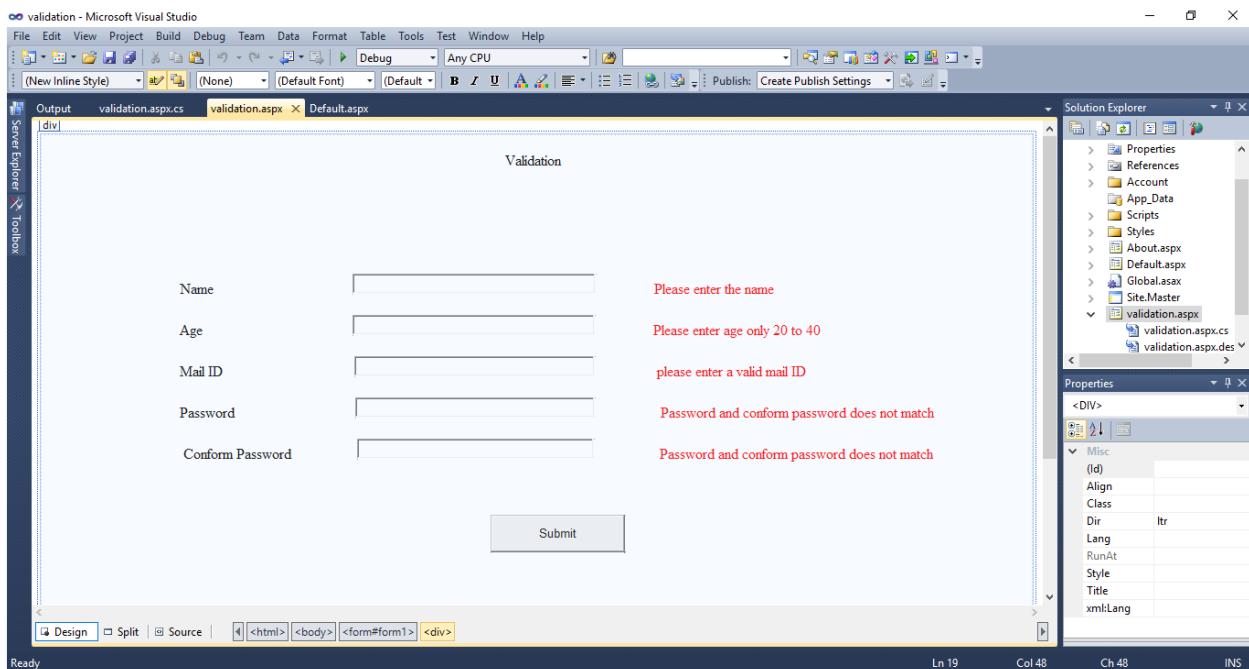
Date:

VALIDATION CONTROLS

Aim

To write a Web Application program to perform validation controls.

Design Window



Source Code

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="validation.aspx.cs"
Inherits="validation.validation" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

    <title></title>

</head>
```



```
<asp:Label ID="Label5" runat="server" Text="Mail ID"></asp:Label>
```

```
<asp:TextBox ID="TextBox4" runat="server" Width="263px"></asp:TextBox>
```

```
<asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server"
```

ErrorMessage="please enter a valid mail ID" ControlToValidate="TextBox4"

```
validationExpression=          "|w+([-.'|w+)*@|w+([-.|w+)*|.|w+([-.]|w+)*"
ForeColor="Red" SetFocusOnError="true" Display="Dynamic">>
```

```
</asp:RegularExpressionValidator>
```



```
<asp:TextBox ID="TextBox5" runat="server" Width="262px"></asp:TextBox>
```

```
<asp:CompareValidator ID="CompareValidator1" runat="server">
```



```
<asp:Button ID="Button1" runat="server" Height="41px" Text="Submit"
```

Width="147px" onclick="Button1_Click" />

</div>

</form>

</body>

</html>

Program Coding

```
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace validation
{
    public partial class validation : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.Write("Submitted Sucessfully");
        }
    }
}
```



Output Window

A screenshot of a web browser window titled "Validation". The page contains five input fields: Name, Age, Mail ID, Password, and Conform Password. The "Name" field is empty and has a red validation message "Please enter the name" to its right. A "Submit" button is located below the input fields.

Name	<input type="text"/>	Please enter the name
Age	<input type="text"/>	
Mail ID	<input type="text"/>	
Password	<input type="text"/>	
Conform Password	<input type="text"/>	

A screenshot of a web browser window titled "Validation". The page contains five input fields: Name, Age, Mail ID, Password, and Conform Password. The "Name" field contains "Mahi" and has a red validation message "Please enter age only 20 to 40" to its right. The "Mail ID" field contains "mahi@gmail.com" and has a red validation message "please enter a valid mail ID" to its right. The "Password" field contains "mahi" and has a red validation message "Password and conform password does not match" to its right. A "Submit" button is located below the input fields.

Name	<input type="text" value="Mahi"/>	Please enter age only 20 to 40
Age	<input type="text" value="50"/>	please enter a valid mail ID
Mail ID	<input type="text" value="mahi@gmail.com"/>	password and conform password does not match
Password	<input type="text" value="mahi"/>	
Conform Password	<input type="text"/>	

Submitted Sucessfully

Validation

Name	<input type="text" value="Mahi"/>
Age	<input type="text" value="20"/>
Mail ID	<input type="text" value="mahi7@gmail.com"/>
Password	<input type="text" value="mahi"/>
Conform Password	<input type="text" value="mahi"/>



EX.NO:17

Date:

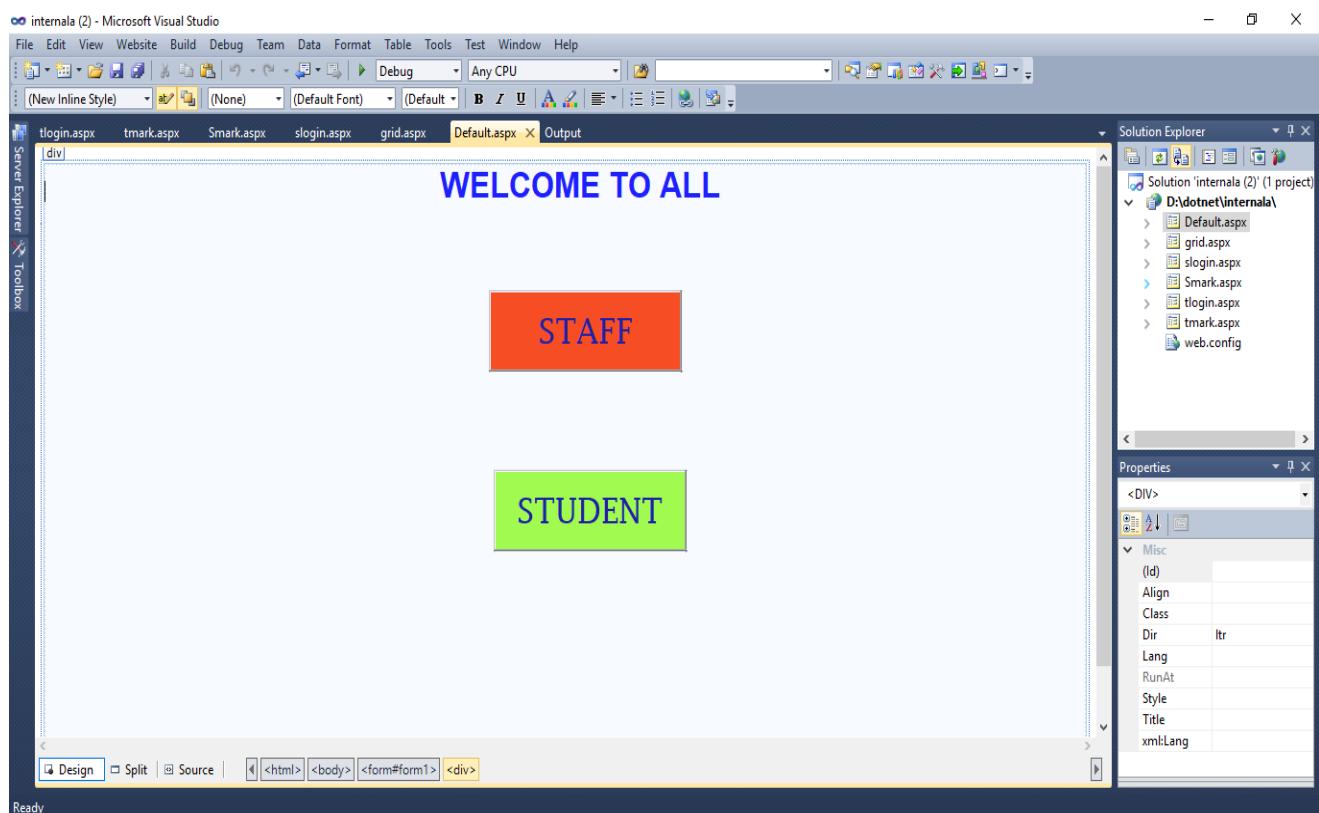
MARK STATEMENT

Aim

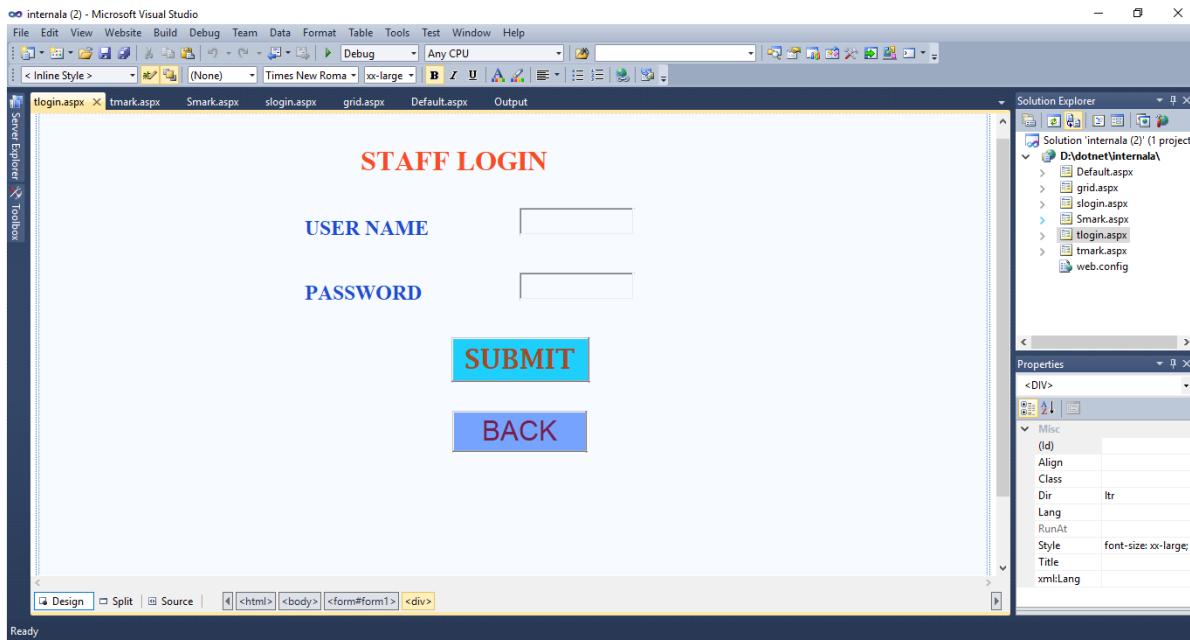
To write a Web Application program to perform internal mark evaluation by staff and the mark view by student using username and password.

Design Window

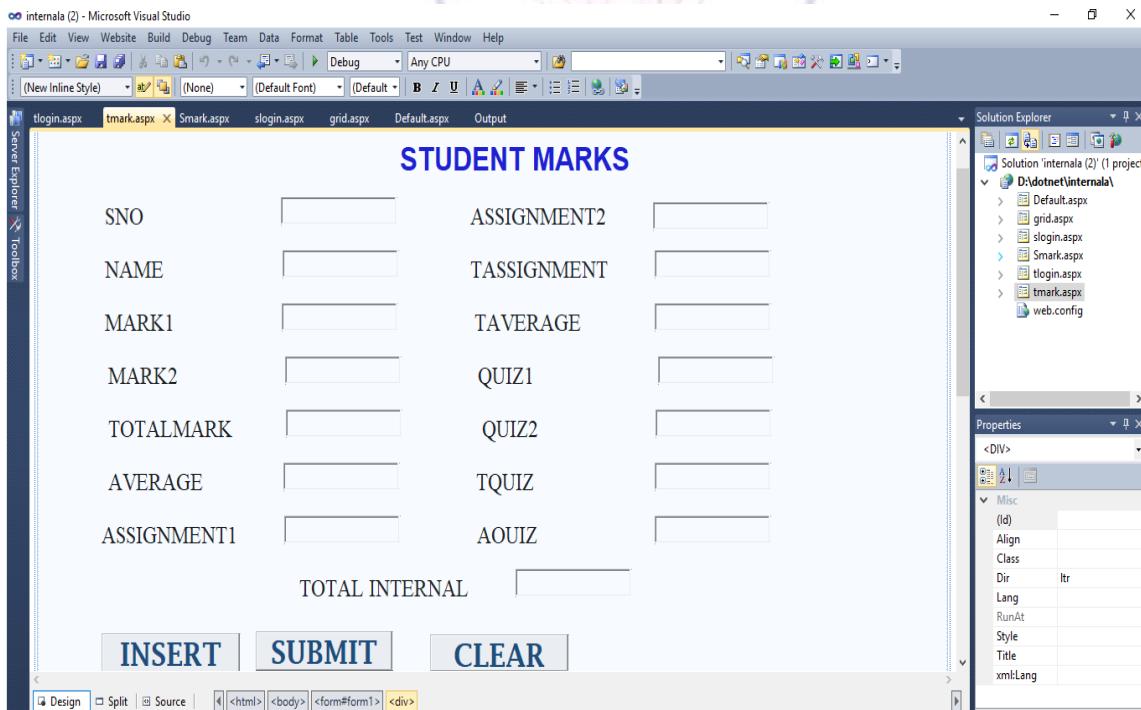
- In this form there is two users one is staff and other is student.



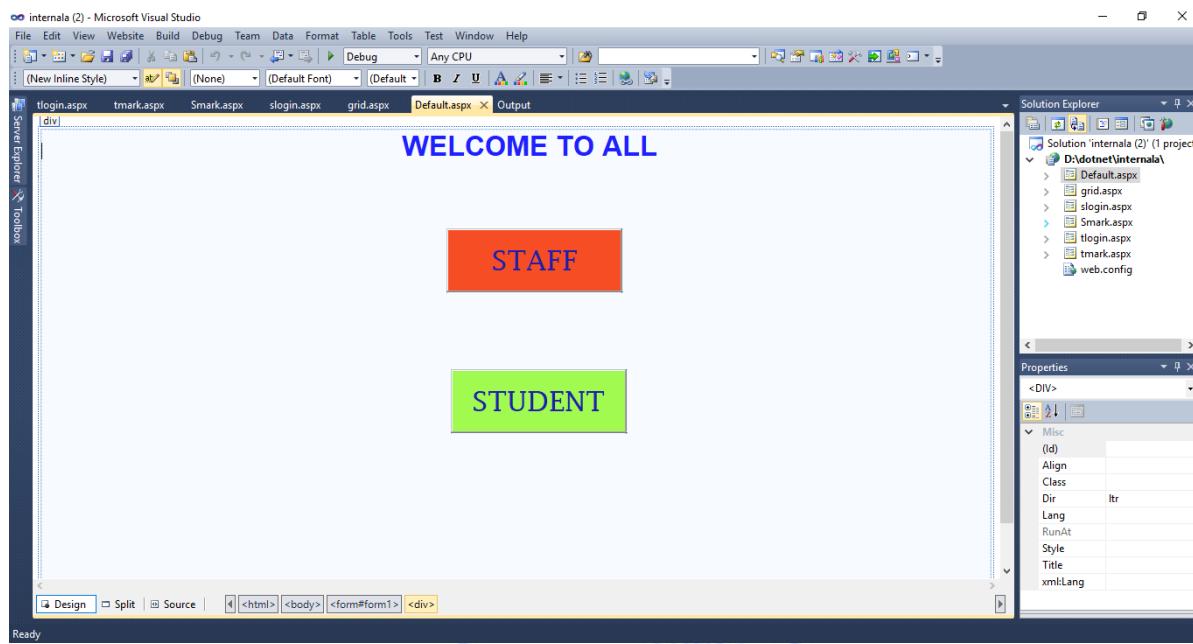
- If we click, staff then display the form of login for staff



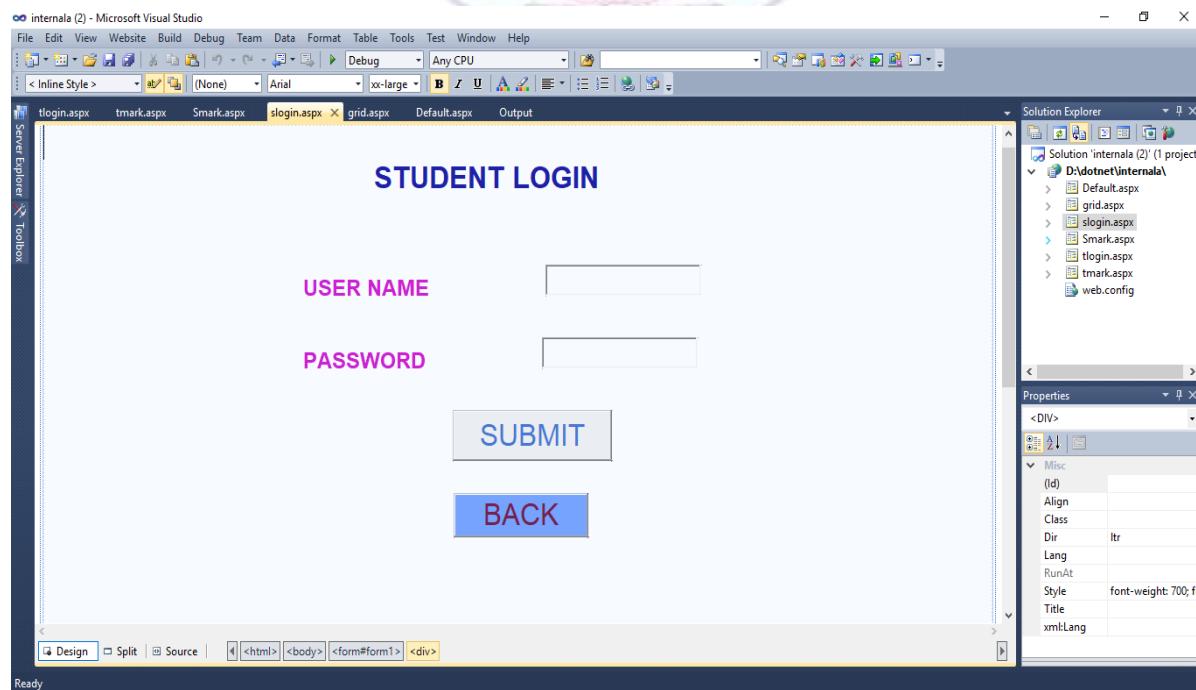
- If username and password match then open student mark
- In this page staff the enter the mark details and submit button calculate the mark and insert is used to store the mark details



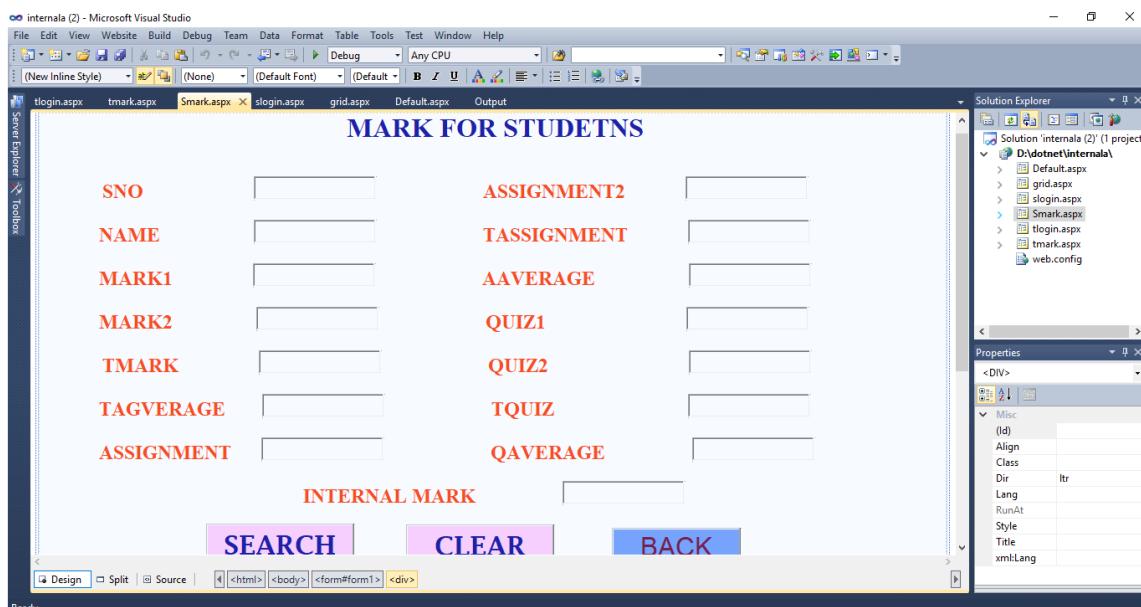
- If staff enter the mark details then student to see the mark details



- If student option click next open login page and then username and password click then open mark display



- In the form student give the ID the click search option then mark are display.



Source Code

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"  
Inherits="_Default" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
```


Staff Login

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="tlogin.aspx.cs"
Inherits="tlogin" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div style="font-size: xx-large; font-family: 'Times New Roman'; font-weight: 700">
```

<asp:Label ID="Label4" runat="server" style="color: #FF3300" Text="STAFF LOGIN"></asp:Label>


```
<asp:Label ID="Label5" runat="server"
    style="font-size: x-large; color: #0033CC" Text="USER NAME"></asp:Label>
    <br />
    <asp:TextBox ID="TextBox1" runat="server"
        style="position: relative; top: 0px; left: 0px; height: 31px; width:
131px"></asp:TextBox>
    <br />
    <asp:Label ID="Label6" runat="server"
        style="font-size: x-large; color: #0033CC" Text="PASSWORD"></asp:Label>
    <br />
    <asp:TextBox ID="TextBox2" runat="server"
        style="position: relative; top: 0px; left: 0px; height: 31px; width:
131px"></asp:TextBox>
    <br />
    <asp:Button ID="Button1" runat="server" onclick="Button1_Click"
        style="position: relative; top: 0px; left: 0px; height: 51px; width: 158px; font-size: xx-
large; font-family: 'Lucida Fax'; color: #993300; font-weight: 700; background-color: #00CCFF"
        Text="SUBMIT" />
    <br />
```

```

 &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<br />
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<asp:Button ID="Button3" runat="server" onclick="Button3_Click"
    style="position: relative; top: -10px; left: -55px; width: 154px; height: 47px; font-size:
    xx-large; font-family: Arial, Helvetica, sans-serif; color: #660033; background-color: #6699FF"
    Text="BACK" />
<br />
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<br />
&nbsp;<br />
<br />
<br />
<br />
</div>
</form>
</body>
</html>

```



Staff mark enter page

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="tmark.aspx.cs"
Inherits="tmark" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

```



```

    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    <asp:TextBox ID="TextBox2" runat="server"
        style="position: relative; top: 0px; left: 2px; height: 27px; width:
138px"></asp:TextBox>
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    <asp:Label ID="Label10" runat="server"
        style="font-family: 'Times New Roman', Times, serif; font-size: x-large"
        Text="TASSIGNMENT"></asp:Label>
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    <asp:TextBox ID="TextBox9" runat="server"
        style="position: relative; top: 0px; left: 0px; height: 27px; width:
138px"></asp:TextBox>
    <br />
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    <br />
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    <asp:Label ID="Label4" runat="server"
        style="font-family: 'Times New Roman', Times, serif; font-size: x-large"
        Text="MARK1"></asp:Label>
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    <asp:TextBox ID="TextBox3" runat="server"
        style="position: relative; top: 0px; left: 0px; height: 27px; width:
138px"></asp:TextBox>
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    <asp:Label ID="Label11" runat="server"
        style="font-family: 'Times New Roman', Times, serif; font-size: x-large"
        Text="TAVERAGE"></asp:Label>
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    <asp:TextBox ID="TextBox10" runat="server"
        style="position: relative; top: 0px; left: 0px; height: 27px; width:
138px"></asp:TextBox>
    <br />
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    <br />
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

```



```
</div>
</form>
</body>
</html>
```

Student Login


```
</div>
</form>
</body>
</html>
```

Student mark for student view the mark

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Smark.aspx.cs"
Inherits="Smark" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

<title></title>

</head>

<body>

```
<form id="form1" runat="server">
```

<div>

```
<asp:Label ID="Label1" runat="server">
```

1

Text="MARK FOR STUDENTS"></asp:Label>


```
<asp:Label ID="Label2" runat="server">
```

Text="SNO"></asp:Label>

```
<asp:TextBox ID="TextBox1" runat="server">
```

></asp:TextBox>


```

<br />
    <asp:Label ID="Label14" runat="server"
        style="font-weight: 700; font-size: x-large; color: #FF3300"
    Text="ASSIGNMENT"></asp:Label>
    <asp:TextBox ID="TextBox13" runat="server"
        style="position: relative; top: 2px; left: 0px; height: 28px; width:
146px"></asp:TextBox>
    <asp:Label ID="Label15" runat="server"
        style="font-weight: 700; font-size: x-large; color: #FF3300"
    Text="QAVERAGE"></asp:Label>
    <asp:TextBox ID="TextBox14" runat="server"
        style="position: relative; top: 2px; left: 0px; height: 28px; width:
146px"></asp:TextBox>
    <br />
    <br />
    <asp:Label ID="Label16" runat="server"
        style="font-weight: 700; font-size: x-large; color: #FF3300"
    Text="INTERNAL MARK"></asp:Label>
    <asp:TextBox ID="TextBox15" runat="server"
        style="position: relative; top: 2px; left: 0px; height: 28px; width:
146px"></asp:TextBox>
    <br />

```



```
</body>
</html>
```

Program Coding

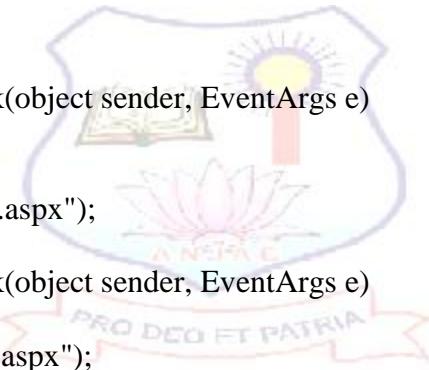
Default page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
        Response.Redirect("slogin.aspx");
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Response.Redirect("tlogin.aspx");
    }
}
```



Staff Login

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.OleDb;

public partial class slogin : System.Web.UI.Page
{
```

```

protected void Page_Load(object sender, EventArgs e)
{
}

OleDbConnection con = new
OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\aba\inter.accdb");
protected void Button1_Click(object sender, EventArgs e)
{
    con.Open();
    OleDbCommand cmd = new OleDbCommand();
    cmd.Connection = con;
    cmd.CommandText = "select * from Slogin where(username='" + TextBox1.Text + "' and
password='" + TextBox2.Text + "')";
    OleDbDataReader r = cmd.ExecuteReader();
    if (r.Read())
    {
        Response.Redirect("Smark.aspx");
    }
    else
    {
        Response.Redirect("wrong");
    }
    con.Close();
}
protected void Button3_Click(object sender, EventArgs e)
{
    Response.Redirect("Default.aspx");
}
}

```

Staff mark enter page

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.OleDb;

public partial class tmark : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
}

```

```

OleDbConnection con = new
OleDbConnection(@ "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\aba\inter.accdb");
OleDbCommand cmd = new OleDbCommand();
protected void Button1_Click(object sender, EventArgs e)
{
    con.Open();
    OleDbCommand cmd = new OleDbCommand();
    cmd.Connection = con;
    cmd.CommandText = "insert into mark values(" + TextBox1.Text + "," + TextBox2.Text
+ "," + TextBox3.Text + "," + TextBox4.Text + "," + TextBox5.Text + "," + TextBox6.Text
+ "," + TextBox7.Text + "," + TextBox8.Text + "," + TextBox9.Text + "," + TextBox10.Text
+ "," + TextBox11.Text + "," + TextBox12.Text + "," + TextBox13.Text + "," +
TextBox14.Text + "," + TextBox15.Text + ")";
    cmd.ExecuteNonQuery();
    con.Close();
}
protected void Button2_Click(object sender, EventArgs e)
{
    double a, b, c, d, s, f, h, g, i, j, k, l, p, q, r, t;
    a = Convert.ToDouble(TextBox3.Text);
    b = Convert.ToDouble(TextBox4.Text);
    c = a + b;
    TextBox5.Text = c.ToString();
    d = c / 2;
    TextBox6.Text = d.ToString();
    s = Convert.ToDouble(TextBox7.Text);
    f = Convert.ToDouble(TextBox8.Text);
    g = s + f;
    TextBox9.Text = g.ToString();
    h = g / 2;
    TextBox10.Text = h.ToString();
    i = Convert.ToDouble(TextBox11.Text);
    j = Convert.ToDouble(TextBox12.Text);
    k = i + j;
    TextBox13.Text = k.ToString();
    l = k / 2;
    TextBox14.Text = l.ToString();
    p = Convert.ToDouble(TextBox6.Text);
    q = Convert.ToDouble(TextBox10.Text);
    r = Convert.ToDouble(TextBox14.Text);
    t = p + q + r;
    TextBox15.Text = Convert.ToString(t);
}

```

```

        }
        protected void Button3_Click(object sender, EventArgs e)
        {
            TextBox1.Text = "";
            TextBox2.Text = "";
            TextBox3.Text = "";
            TextBox4.Text = "";
            TextBox5.Text = "";
            TextBox6.Text = "";
            TextBox7.Text = "";
            TextBox8.Text = "";
            TextBox9.Text = "";
            TextBox10.Text = "";
            TextBox11.Text = "";
            TextBox12.Text = "";
            TextBox13.Text = "";
            TextBox14.Text = "";

        }
        protected void Button5_Click(object sender, EventArgs e)
        {
            Response.Redirect("tlogin.aspx");
        }
    }
}

```

Student Login



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.OleDb;

public partial class slogin : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    OleDbConnection con = new
    OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\aba\inter.accdb");
    protected void Button1_Click(object sender, EventArgs e)
    {

```

```

con.Open();
OleDbCommand cmd = new OleDbCommand();
cmd.Connection = con;
cmd.CommandText = "select * from Slogin where(username=''" + TextBox1.Text + "' and
password=''" + TextBox2.Text + "'')";
OleDbDataReader r = cmd.ExecuteReader();
if (r.Read())
{
    Response.Redirect("Smark.aspx");
}
else
{
    Response.Redirect("wrong");
}
con.Close();
}

protected void Button3_Click(object sender, EventArgs e)
{
    Response.Redirect("Default.aspx");
}
}

```

Student mark view page



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.OleDb;

public partial class Smark : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    OleDbConnection con = new
    OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=D:\aba\inter.accdb");
    protected void Button2_Click(object sender, EventArgs e)
    {
        TextBox1.Text = "";
        TextBox2.Text = "";
        TextBox3.Text = "";
    }
}

```

```

    TextBox4.Text = "";
    TextBox5.Text = "";
    TextBox6.Text = "";
    TextBox7.Text = "";
    TextBox8.Text = "";
    TextBox9.Text = "";
    TextBox10.Text = "";
    TextBox11.Text = "";
    TextBox12.Text = "";
    TextBox13.Text = "";
    TextBox14.Text = "";

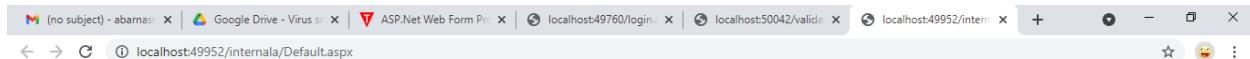
}

protected void Button1_Click(object sender, EventArgs e)
{
    con.Open();
    OleDbCommand cmd = new OleDbCommand();
    cmd.Connection = con;
    cmd.CommandText = "select * from mark where sid=" + TextBox1.Text + "";
    OleDbDataReader dr = cmd.ExecuteReader();
    while (dr.Read())
    {
        TextBox3.Text=dr["sname"].ToString();
        TextBox5.Text = dr["m1"].ToString();
        TextBox7.Text = dr["m2"].ToString();
        TextBox9.Text = dr["mt"].ToString();
        TextBox11.Text = dr["ma"].ToString();
        TextBox13.Text = dr["a1"].ToString();
        TextBox2.Text = dr["a2"].ToString();
        TextBox4.Text = dr["at"].ToString();
        TextBox6.Text = dr["aa"].ToString();
        TextBox8.Text = dr["q1"].ToString();
        TextBox10.Text = dr["q2"].ToString();
        TextBox12.Text = dr["qt"].ToString();
        TextBox14.Text = dr["qa"].ToString();
        TextBox15.Text = dr["it"].ToString();
    }
    con.Close();
}

protected void Button3_Click(object sender, EventArgs e)
{
    Response.Redirect("slogin.aspx");
}
}

```

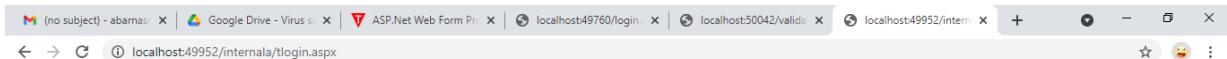
Output Window



WELCOME TO ALL

STAFF

STUDENT



STAFF LOGIN

USER NAME

PASSWORD

SUBMIT

BACK

STUDENT MARKS

SNO	<input type="text"/>	ASSIGNMENT2	<input type="text"/>
NAME	<input type="text"/>	TASSIGNMENT	<input type="text"/>
MARK1	<input type="text"/>	TAVERAGE	<input type="text"/>
MARK2	<input type="text"/>	QUIZ1	<input type="text"/>
TOTALMARK	<input type="text"/>	QUIZ2	<input type="text"/>
AVERAGE	<input type="text"/>	TQUIZ	<input type="text"/>
ASSIGNMENT1	<input type="text"/>	AOUIZ	<input type="text"/>
TOTAL INTERNAL		<input type="text"/>	

INSERT **SUBMIT** **CLEAR**

BACK



STUDENT MARKS

SNO	<input type="text" value="s2"/>	ASSIGNMENT2	<input type="text" value="8.5"/>
NAME	<input type="text" value="yyy"/>	TASSIGNMENT	<input type="text"/>
MARK1	<input type="text" value="89"/>	TAVERAGE	<input type="text"/>
MARK2	<input type="text" value="97"/>	QUIZ1	<input type="text" value="4"/>
TOTALMARK	<input type="text"/>	QUIZ2	<input type="text" value="4.5"/>
AVERAGE	<input type="text"/>	TQUIZ	<input type="text"/>
ASSIGNMENT1	<input type="text" value="10"/>	AOUIZ	<input type="text"/>
TOTAL INTERNAL		<input type="text"/>	

INSERT **SUBMIT** **CLEAR**

BACK

M (no subject) - abarnas | Google Drive - Virus | ASP.Net Web Form Proj | localhost:49760/login | localhost:50042/valid | localhost:49952/internal/tmark.aspx | localhost:49952/internal | + | - | X | ← | → | C | i | localhost:49952/internal/tmark.aspx | ☆ | : |

STUDENT MARKS

SNO	s2	ASSIGNMENT2	8.5
NAME	yyy	TASSIGNMENT	18.5
MARK1	89	TAVERAGE	9.25
MARK2	97	QUIZ1	4
TOTALMARK	186	QUIZ2	4.5
AVERAGE	93	TQUIZ	8.5
ASSIGNMENT1	10	AOUIZ	4.25
TOTAL INTERNAL		106.5	

INSERT **SUBMIT** **CLEAR**

BACK



M (no subject) - abarnas | Google Drive - Virus | ASP.Net Web Form Proj | localhost:49760/login | localhost:50042/valid | localhost:49952/internal | + | - | X | ← | → | C | i | localhost:49952/internal | ☆ | : |

WELCOME TO ALL

STAFF

STUDENT

STUDENT LOGIN

USER NAME

PASSWORD

SUBMIT

BACK



MARK FOR STUDENTS

SNO	<input type="text" value="s1"/>	ASSIGNMENT2	<input type="text"/>
NAME	<input type="text"/>	TASSIGNMENT	<input type="text"/>
MARK1	<input type="text"/>	AAVERAGE	<input type="text"/>
MARK2	<input type="text"/>	QUIZ1	<input type="text"/>
TMARK	<input type="text"/>	QUIZ2	<input type="text"/>
TAGVERAGE	<input type="text"/>	TQUIZ	<input type="text"/>
ASSIGNMENT	<input type="text"/>	QAVERAGE	<input type="text"/>

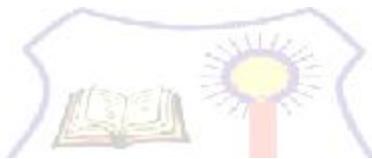
INTERNAL MARK

SEARCH **CLEAR** **BACK**

inter : Database (Access 2007) - Microsoft Access

Datasheet View

sid	sname	m1	m2	mt	ma	a1	a2	at	aa	q1	q2
s1	xxx	90	80	170	85	10	10	20	9.75	5	4
s2	YY	89	97	186	93	10	8	18	9.25	4	4



inter : Database (Access 2007) - Microsoft Access

Datasheet View

username	password
s1	s2
*	

inter : Database (Access 2007) - Microsoft Access

The screenshot shows the Microsoft Access 2007 interface with the 'Datasheet' tab selected. On the left, the 'All Tables' pane lists four tables: 'tlogin', 'mark', 'slogin', and 'slogon'. The 'tlogin' table is currently selected and displayed in the main area. It contains two records:

username	password
t1	t1
t2	t2

Below the table, the status bar shows 'Record: 1 of 2'.

