Velagapudi Ramakrishna Siddhartha Engineering College

Program Elective - 1 Course

Dot Net Technologies Lab

Code : 20IT5404B

INDEX

Sl. No	Exp No.	Date	Name of the Experiment	Page No.	Marks		Date of Submission Sign
					Remarks	Viva	
1	1	17-8-22	Program on Classes and Objects				
	2		Program On Properties and Indexers				
2		02-09-22	Program On Arrays, Jagged				
3		14-09-22	Program on Getting and Display Student Details and Computing Average				
4	1	14-09-22	Program On Parms Keyword				
	2		Program On Structures				
5		12-10-22	Program On Delegates				
6		19-10-22	Program On Interfaces				
7	1	02-11-22	Programs on Abstract Classes				
	2		Programs On Inheritances				
8		09-11-22	Interaction With Windows Form and Database Connection				
9		17-11-22	Application On ASP.Net Razor pages				
10		24-11-22	Case Study Stock Management Application				
11		01-12-22	Case Study Stock Management Application				

Week – 1

Aim : Create C# class Employee with the fields eno, ename, address, designation, mobile number, salary, city, pincode. Create necessary methods to read the information and display the information. The employer contains 10 employees. The employer wants to know the total salary paid to its employees. As a programmer how do you suggest a solution to his problem?

Program :

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.Cryptography.X509Certificates;

using System.Text;

using System.Threading.Tasks;


namespace Employee_1

{

   public class Program

   {

      public decimal total = 0;

      public static void Main(string[] args)

      {
```

```
        decimal T = 0;
        Employee[] obj = new Employee[10];

        for(int i = 0; i < 3; i++)
        {
            obj[i] = new Employee();
            obj[i].SetStudentDetails();
            Console.WriteLine();
            obj[i].GetStudentDetails();

            decimal t = obj[i].salary;
            T = T + t;
        }

        Console.WriteLine("Total Salary of all Employee's is : " + T);

    }
}

class Employee
{
```

```
public int eno, pincode;
public string ename, address, designation, city;
public decimal salary;
public double mobile;

public void SetStudentDetails()
{
    Console.Write("Enter Employee number : ");
    eno = Convert.ToInt32(Console.ReadLine());

    Console.Write("Enter Employee Name : ");
    ename = Console.ReadLine();

    Console.Write("Enter Employee Location Address : ");
    address = Console.ReadLine();

    Console.Write("Enter Employee Designation : ");
    designation = Console.ReadLine();

    Console.Write("Enter Employee's Mobile number : ");
    mobile = Convert.ToDouble(Console.ReadLine());
```

```
        Console.Write("Enter Employee Salary : ");
        salary = Convert.ToDecimal(Console.ReadLine());
        Console.Write("Enter Employee Located City's Name : ");
        city = Console.ReadLine();
        Console.Write("Enter Area pincode : ");
        pincode = Convert.ToInt32(Console.ReadLine());
    }
    public void GetStudentDetails()
    {
        Console.WriteLine("Employee Number : " + eno);
        Console.WriteLine("Employee's Name : " + ename);
        Console.WriteLine("Employee Address : " + address);
        Console.WriteLine("Employee Designation : " + designation);
        Console.WriteLine("Employee's Mobile Number : " + mobile);
        Console.WriteLine("Employee's Salary : " + salary);
        Console.WriteLine("Employee City : " + city);
        Console.WriteLine("Employee Area Pincode : " + pincode);
    }
  }
}
```

Output :

C:\Windows\system32\cmd.exe

```
Enter Employee number : 100
Enter Employee Name : sai
Enter Employee Location Address : bhavanipuram
Enter Employee Designation : student
Enter Employee's Mobile number : 8331052556
Enter Employee Salary : 85000
Enter Employee Located City's Name : vijayawada
Enter Area pincode : 520012

Employee Number : 100
Employee's Name : sai
Employee Address : bhavanipuram
Employee Designation : student
Employee's Mobile Number : 8331052556
Employee's Salary : 85000
Employee City : vijayawada
Employee Area Pincode : 520012

Enter Employee number : 99
Enter Employee Name : rizwanullah
Enter Employee Location Address : times_hospital
Enter Employee Designation : student
Enter Employee's Mobile number : 6457634534
Enter Employee Salary : 95000
Enter Employee Located City's Name : vijayawada
Enter Area pincode : 520001
```

```
Employee Number : 91
Employee's Name : mounav
Employee Address : Poranki
Employee Designation : student
Employee's Mobile Number : 7825763256
Employee's Salary : 75000
Employee City : vijayawada
Employee Area Pincode : 520001

Total Salary of all Employee's is : 255000
Press any key to continue . . .
```

Result : Sucessfully Executed the program.

Aim : Create a C# class to read the book information to a library which includes booknumber, name, pages, issn number, price, year_of_publication. Use C# Properties to store the information into these variables and retrieve the information.

Program :

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Properties_1

{

   internal class Program

   {

      public static void Main(string[] args)

      {

         Person obj = new Person();

         Console.Write("Enter Book Number : ");

         obj.book_no = Convert.ToInt32(Console.ReadLine());
```

```
        Console.Write("Enter Book Name : ");
        obj.name = Console.ReadLine();
        Console.Write("Enter No.of Pages in Book : ");
        obj.pages = Convert.ToInt32(Console.ReadLine());
        Console.Write("Enter Book ISSN Number : ");
        obj.issn = Convert.ToInt32(Console.ReadLine());
        Console.Write("Enter Book Price : ");
        obj.price = Convert.ToInt32(Console.ReadLine());
        Console.Write("Enter Book Publishe Year : ");
        obj.pub = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine();

        Console.WriteLine("Book Number : " + obj.book_no);
        Console.WriteLine("Book Name : " + obj.name);
        Console.WriteLine("No.of Pages in That Book : " + obj.pages);
        Console.WriteLine("Book ISSN Number : " + obj.issn);
        Console.WriteLine("Book Price : " + obj.price);
        Console.WriteLine("Year of Published : " + obj.pub);
    }
}
```

```
    class Person
    {
        public int book_no { get; set; }
        public string name { get; set; }
        public int pages { get; set; }
        public int issn { get; set; }
        public int price { get; set; }
        public int pub { get; set; }
    }
}
```

Output :

```
C:\Windows\system32\cmd.exe
Enter Book Number : 10
Enter Book Name : Csharp
Enter No.of Pages in Book : 1024
Enter Book ISSN Number : 87
Enter Book Price : 500
Enter Book Publishe Year : 2019

Book Number : 10
Book Name : Csharp
No.of Pages in That Book : 1024
Book ISSN Number : 87
Book Price : 500
Year of Published : 2019
```

Result : Sucessfully Executed the Program.

Week – 2

Aim : Write a C# program of jagged array which declares, initializes and traverse jagged arrays.

Program :

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Jagged_Arrays

{

   internal class Program

   {

      static void Main(string[] args)

      {

         int[][] arr = new int[2][]; // Declare the array

         arr[0] = new int[] { 11, 21, 56, 78 };// Initialize the array

         arr[1] = new int[] { 42, 61, 37, 41, 59, 63 };

         Console.WriteLine("Jagged Array : ");

         // Traverse array elements

         for (int i = 0; i < arr.Length; i++)
```

```
        {
            for (int j = 0; j < arr[i].Length; j++)
            {
                System.Console.Write(arr[i][j] + " ");
            }
            System.Console.WriteLine();
        }
    }
  }
}
```

Output :

Result : Sucessfully Executed the program.

Week – 3

Aim : Create a class Student with the filed sno, sname, marks in three subjects. Create a method SetStudentDetails() which will read the information from the student. Create another method GetStudentDetails() which will display the information. Also, another method to compute the average mark of the student in three subject. Display the information.

Program :

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Student_Class

{

   public class Program

   {

      static void Main(string[] args)

      {

         Student obj = new Student();

         obj.SetStudentDetails();

         obj.average();
```

```
        Console.WriteLine();
        obj.GetStudentDetails();
    }
}
class Student
{
    public int sno;
    public string sname;
    public Decimal m1, m2, m3;
    public void SetStudentDetails()
    {
        Console.Write("Enter the Student Number : ");
        sno = Convert.ToInt32(Console.ReadLine());
        Console.Write("Enter the Student Name : ");
        sname = Console.ReadLine();
        Console.Write("Enter the Student marks in Subject 1: ");
        m1 = Convert.ToDecimal(Console.ReadLine());
        Console.Write("Enter the Student marks in Subject 2: ");
        m2 = Convert.ToDecimal(Console.ReadLine());
        Console.Write("Enter the Student marks in Subject 3: ");
        m3 = Convert.ToDecimal(Console.ReadLine());
```

```
        }
        public void GetStudentDetails()
        {
            Console.WriteLine("Student Number is : " + sno);
            Console.WriteLine("Student Name : " + sname);
            Console.WriteLine("Student Marks : " + m1 + " " + m2 + " " + m3);
        }
        public void average()
        {
            Console.WriteLine("Average of 3 Subjects : " + ((m1 + m2 + m3) / 3));
        }
    }
}
```

Output :

```
C:\Windows\system32\cmd.exe
Enter the Student Number : 92
Enter the Student Name : satya
Enter the Student marks in Subject 1: 90
Enter the Student marks in Subject 2: 95
Enter the Student marks in Subject 3: 85
Average of 3 Subjects : 90

Student Number is : 92
Student Name : satya
Student Marks : 90 95 85
Press any key to continue . . .
```

Result : Sucessfully Executed the Program .

Week – 4

Aim : Write a C# program which takes variable number of parameters i.e using params.

Program :

```
using System;
namespace Params_Keyword
{
   internal class Program
   {
      public void Show(params int[] val) // Params Paramater
      {
         Console.Write("List of all values : ");
         for (int i = 0; i < val.Length; i++)
         {
            Console.Write(val[i] + " ");
         }
         Console.ReadLine();
      }
      static void Main(string[] args)
      {
```

```
        Program p = new Program(); // Creating Object
        p.Show(2, 4, 6, 8, 10, 12, 14); // Passing arguments of variable length
        p.Show(1, 2, 3, 4, 5);
    }
  }
}
```

Output :

```
C:\Windows\system32\cmd.exe
List of all values : 2 4 6 8 10 12 14
List of all values : 1 2 3 4 5
Press any key to continue . . .
```

Result : Sucessfully executed the program.

Aim : Write a C# program to create a struct Rectangle which has two data members width and height, where we are using constructor to initialize data and method to calculate area of rectangle.

Program :

```
using System;

namespace Structures_2

{

   internal class Program

   {

      public static void Main(string[] args)

      {

         Rectangle r = new Rectangle(4,5);

         Console.WriteLine("Area of the Rectangle with parameterized values");

         r.Area_Rectangle();

         Console.WriteLine();

         Console.WriteLine("Area of the Rectangle with Customized values : ");

         r.width = 10;

         r.height = 20;

         r.Area_Rectangle();
```

```
        }
    }
    public struct Rectangle
    {
        public int width, height;
        public Rectangle(int w, int h)
        {
            this.width = w;
            this.height = h;
        }
        public void Area_Rectangle()
        {
            Console.WriteLine("Area of the Rectangle : " + (width * height));
        }
    }
}
```

Output :

```
C:\Windows\system32\cmd.exe
Area of the Rectangle with parameterized values
Area of the Rectangle : 20

Area of the Rectangle with Customized values :
Area of the Rectangle : 200
Press any key to continue . . .
```

Result : Sucessfully Executed the Program.

Week – 5

Aim : Write a C# program on Multicast Delegates.

Program :

```
using System;
namespace Multicast_Delegates
{
   internal class Program
   {
      public delegate void delmethod(int x, int y);
      public void plus_Method1(int x, int y)
      {
         Console.Write("You are in plus_Method");
         Console.WriteLine(x + y);
      }

      public void subtract_Method2(int x, int y)
      {
         Console.Write("You are in subtract_Method");
         Console.WriteLine(x - y);
```

```
        }
        static void Main(string[] args)
        {
            Program obj = new Program();
            delmethod del = new delmethod(obj.plus_Method1);
            // Here we have multicast
            del += new delmethod(obj.subtract_Method2);
            // plus_Method1 and subtract_Method2 are called
            del(50, 10);
            Console.WriteLine();
            //Here again we have multicast
            del -= new delmethod(obj.plus_Method1);
            //Only subtract_Method2 is called
            del(20, 10);
            Console.ReadLine();
        }
    }
}
```

Output :

```
C:\Windows\system32\cmd.exe
You are in plus_Method60
You are in subtract_Method40

You are in subtract_Method10

Press any key to continue . . .
```

Result : Sucessfully Executed the program.

Week – 6

Aim : Write a C# program to create an interface Drawable which has draw() method. Its implementation is provided by two classes: Rectangle and Circle.

Program :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Interfaces
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Drawable d;
            d = new Rectangle();
            d.draw();
            d = new Circle();
            d.draw();
```

```
    }
}
public interface Drawable
{
    void draw();
}
public class Rectangle : Drawable
{
    public void draw()
    {
        Console.WriteLine("drawing rectangle...");
    }
}
public class Circle : Drawable
{
    public void draw()
    {
        Console.WriteLine("drawing circle...");
    }
}
```

}

Output :

Result : Sucessfully Executed the Program.

Week – 7

Aim : Create a Class Polygon with three variables l,b,h. Create a parameterised method to set the values to l,b,h. Create an abstract method compute().

Program :

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Abstract_Class

{

   internal class Program

   {

      static void Main(string[] args)

      {

         Cuboid c1 = new Cuboid(5, 10, 15);

         c1.Compute();
```

```
        Cube c2 = new Cube(2, 8, 4);
        c2.Compute();
    }
}

abstract class Polygon
{
    public int l, b, h;
    public Polygon(int l, int b, int h)
    {
        this.l = l;
        this.b = b;
        this.h = h;
    }

    public abstract void Compute();
}

class Cuboid : Polygon
{
    public Cuboid(int l, int b, int h) : base(l, b, h)
```

```
    {
        //base(l, b, h);
    }

    public override void Compute()
    {
        Console.WriteLine("Volume of Cuboid : " + (l * b * h));
    }
}

class Cube : Polygon
{
    public Cube(int l, int b, int h) : base(l, b, h)
    {
        //base(l, b, h);
    }
    public override void Compute()
    {
        Console.WriteLine("Area of the Cube : " + (l * b * h));
    }
}
```

}

Output :

C:\Windows\system32\cmd.exe

```
Volume of Cuboid : 750
Area of the Cube : 64
Press any key to continue . . .
```

Result : Sucessfully executed the program.

Aim : Implement the following inheritance :

Program :

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace Bank_Console

{

   internal class Program

   {

      static void Main(string[] args)
```

```
    {
        CheckAccount ca = new CheckAccount();
        Console.WriteLine("Initial Amount in Account : " + ca.amount);

        ca.ProcessCheck(2000);
        Console.WriteLine("Total Amount after WithDraw : " + ca.amount);

        ca.Deposit(6000);
        Console.WriteLine("Total amount after Deposited : " + ca.amount);

        SavingsAccount sa = new SavingsAccount();
        sa.DepositMonthlyInterest();
        Console.WriteLine(" Amount with Interest : " + sa.amount);

    }
}

class BankAccount
{
    public string owner;
    public float balance;
```

```
    public float amount = 10000;
    public void Deposit(float new_amount)
    {
        this.amount = this.amount + new_amount;
    }

    public void WithDrawl(float old_amount)
    {
        this.amount = this.amount - old_amount;
    }
}
class CheckAccount : BankAccount
{
    public float InsufficentFee;

    public void ProcessCheck(float check_amount)
    {
        if (check_amount > base.amount)
        {
            Console.WriteLine(" You Don't Have Sufficent Amount in Account");
        }
```

```
        else
        {
            base.WithDrawl(check_amount);
        }
    }
}
class SavingsAccount : BankAccount
{
    public float AnnualInterest = 0.9f;


    public void DepositMonthlyInterest()
    {
        base.amount = base.amount * AnnualInterest;
    }
}
}
```

Output :

C:\Windows\system32\cmd.exe

```
Initial Amount in Account : 10000
Total Amount after WithDraw : 8000
Total amount after Deposited : 14000
 Amount with Interest : 9000
Press any key to continue . . .
```

Result : Sucessfully Executed the program.

Week – 8

Aim : Create an windows form application which interacts with Database and performs the Employee CRUD Operations.

Procedure :

1. First Create a table with in the data base with below query

Query :

```
CREATE TABLE EmployeeData
(
   [Emp_ID] INT NOT NULL PRIMARY KEY DEFAULT 0,
   [Emp_Name] VARCHAR(50) NULL,
   [Age] INT NULL,
   [Gender] VARCHAR(50) NULL,
   [Designation] VARCHAR(50) NULL,
   [Dept_ID] INT NULL,
   [Dept_Name] VARCHAR(50) NULL,
   [Location] VARCHAR(50) NULL,
   [Work_Project] VARCHAR(50) NULL,
   [Company] VARCHAR(50) NULL,
   [Salary] FLOAT NULL
)
```

2. After creating the table then save it with name EmployeeData.

3. Now, Create a windows form in the visual studio just like the below one

4. After creating the windows form now, write the code for each button
5. The code is just below down there.

Code For Each button :

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```

```
using System.Windows.Forms;

using System.Data.SqlClient;

namespace Employee_CRUD

{

    public partial class Form1 : Form

    {

        public Form1()

        {

            InitializeComponent();

        }

        private void button1_Click(object sender, EventArgs e)

        {

            //Insert Button

            SqlConnection con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\SHREEE\\OneDrive\\Documents\\My
DataBase.mdf;Integrated Security=True;Connect Timeout=30");

            con.Open();

            SqlCommand cmd = new SqlCommand("insert into EmployeeData
values(@Emp_ID,@Emp_Name,@Age,@Gender,@Designation,@Dept_ID,@Dept_Name,@Location,@Work_Pro
ject,@Company,@Salary)",con);



            cmd.Parameters.AddWithValue("@Emp_ID", int.Parse(textBox1.Text));

            cmd.Parameters.AddWithValue("@Emp_Name", textBox2.Text);

            cmd.Parameters.AddWithValue("@Age", int.Parse(textBox3.Text));

            cmd.Parameters.AddWithValue("@Gender", textBox4.Text);

            cmd.Parameters.AddWithValue("@Designation", textBox5.Text);

            cmd.Parameters.AddWithValue("@Dept_ID", int.Parse(textBox6.Text));

            cmd.Parameters.AddWithValue("@Dept_Name", textBox7.Text);

            cmd.Parameters.AddWithValue("@Location", textBox8.Text);

            cmd.Parameters.AddWithValue("@Work_Project", textBox9.Text);

            cmd.Parameters.AddWithValue("@Company", textBox10.Text);
```

```
            cmd.Parameters.AddWithValue("@Salary", double.Parse(textBox11.Text));



            cmd.ExecuteNonQuery();

            con.Close();

            MessageBox.Show(" Sucessfully Inserted the Data into Table.. ");

        }

        private void button4_Click(object sender, EventArgs e)

        {

            // Update Button

            SqlConnection con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\SHREEE\\OneDrive\\Documents\\My
DataBase.mdf;Integrated Security=True;Connect Timeout=30");

            con.Open();

            SqlCommand cmd = new SqlCommand();

            cmd.Connection = con;

            cmd.CommandText = "Update EmployeeData set Emp_Name='" + textBox2.Text + "',
Age='" + textBox3.Text + "', Gender='" + textBox4.Text + "', Designation='" +
textBox5.Text + "', Dept_ID='" + textBox6.Text + "', Dept_Name='" + textBox7.Text + "',
Location='" + textBox8.Text + "', Work_Project='" + textBox9.Text + "', Company='" +
textBox10.Text + "', Salary='" + textBox11.Text + "' where Emp_ID='" + textBox1.Text +
"'";

            cmd.ExecuteNonQuery();

            con.Close();

            MessageBox.Show(" Sucessfully Updated the Data into Table");



        }

        private void button2_Click(object sender, EventArgs e)

        {

            // Delete Button

            SqlConnection con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\SHREEE\\OneDrive\\Documents\\My
DataBase.mdf;Integrated Security=True;Connect Timeout=30");

            con.Open();
```

```
            SqlCommand cmd = new SqlCommand("Delete EmployeeData where Emp_ID = @Emp_ID",
con);

            cmd.Parameters.AddWithValue("@Emp_ID", int.Parse(textBox1.Text));

            cmd.ExecuteNonQuery();

            con.Close();

            MessageBox.Show(" Sucessfully Deleted the Data into Table.. ");

        }

        private void button3_Click(object sender, EventArgs e)

        {

            // Display Button

            SqlConnection con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\SHREEE\\OneDrive\\Documents\\My
DataBase.mdf;Integrated Security=True;Connect Timeout=30");

            con.Open();

            SqlCommand cmd = new SqlCommand("Select * from EmployeeData",con);

            SqlDataAdapter da = new SqlDataAdapter(cmd);

            DataTable dt = new DataTable();

            da.Fill(dt);

            dataGridView1.DataSource = dt;

        }

        private void button5_Click(object sender, EventArgs e)

        {

            // Exit Button

            this.Close();

        }

        private void button6_Click(object sender, EventArgs e)

        {

            // Search Button
```

```
            SqlConnection con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\SHREEE\\OneDrive\\Documents\\My
DataBase.mdf;Integrated Security=True;Connect Timeout=30");



            con.Open();

            SqlCommand cmd = new SqlCommand("Select * from EmployeeData where
Emp_ID=@Emp_ID", con);

            cmd.Parameters.AddWithValue("Emp_ID", int.Parse(textBox1.Text));

            SqlDataAdapter da = new SqlDataAdapter(cmd);

            DataTable dt = new DataTable();

            da.Fill(dt);

            dataGridView1.DataSource = dt;

        }

    }

}
```

6. **Make Sure That Use the Connection String of Your Database.**
7. Save All Changes.
8. Now, Finally You can Run the Windows Form

Windows Form Outputs :

Insert Query :

Modify Query :

Delete Query :

Employee
Employee ID :
2345
Employee Name :
Mounav kolluri
Age :
20
Gender :
Male
Designation :
Manager
Department ID :
110
Dept Name :
Information Technology
Location :
Kanuru
Working Project :
Eye Cataract Detection
Company :
VRSEC
Salary :
45000.00
Employee DataBase
Emp_ID
Emp_Name
Age
Gender
1
Ajay
22
Male
2
Rizwan
19
Male
4
M.J.N.V.Sai
20
Male
5
Srinivas
150
Male
999
Mounav
20
Male
2345
Mounav kolluri
20
Male
Insert
Delete
Display
Modify
Search
Exit

Search Query :

Display Query :

When we click on that button it displays the table from the database

Exit :

When we click on the exit button it exits the total application.

Result : Sucessfully Executed the Windows Form Application.

Week – 9

Aim : Design an Asp.Net Core Razor pages Student register application which Interacts with razor pages and Database connection.

Procedure :

1. First we will create an database with a name studentdatabase and this database contains a table with a name called " StudentRegistration "

2. After creating the database and table now click on new project and select " ASP.NET Core Web App "

3. Click on next and give the project name and set the destination path.

4. Then click on again next set the highest framework and check the https

5. After completing all your home page will be like below one.

6. Now right click on the project name in the solution explorer and select " add an item " and then select the new folder option and give it as a name " Models "

7. Right click on the models and then add a class component file and save it with an name "StudentRegistration.cs"

8. a sample pic is below down is there

9. Now write a program on "StudentRegistration.cs" file which declares the attributes in a table as an properties in the class with get and set methods.

10. Now right click on the " Models Folder " and add an razor page .

11. Now choose the " Razor pages using Enity Framework (CRUD) option.

12. In model class select the student registration option

13. Now click on the " plus button " and then add data context and then save all .

In Solution Explorer " /pages/Create.cshtml " file Add this small piece of code

```
<div class="form-group">
                <label asp-for="StudentRegistration.Stuid" class="control-label"></label>
                <input asp-for="StudentRegistration.Stuid" class="form-control" />
                <span asp-validation-for="StudentRegistration.Stuid" class="text-
danger"> </span>
</div>
```

In Appsettings.json file change the Database Connection string.

Final Outputs :

CrudOperations_Razor Home Privacy

VRSEC Student Registration

Create New

Stuname	Email	Salary	
Prabhuram	prabhu@gmail.com	15000	Edit | Details | Delete
Rizwanullah	208w1a1299@vrsec.ac.in	5	Edit | Details | Delete
M J N V Sai	208w1a12a0@vrsec.ac.in	95000	Edit | Details | Delete

© 2022 - CrudOperations_Razor - Privacy

CrudOperations_Razor Home Privacy

Create

StudentRegistration

Stuid

4

Stuname

Kolluri Mounav

Email

291@gmail.com

Salary

99999

Create

Back to List

© 2022 - CrudOperations_Razor - Privacy

CrudOperations_Razor Home Privacy

VRSEC Student Registration

Create New

Stuname	Email	Salary	
Prabhuram	prabhu@gmail.com	15000	Edit | Details | Delete
Rizwanullah	208w1a1299@vrsec.ac.in	5	Edit | Details | Delete
M J N V Sai	208w1a12a0@vrsec.ac.in	95000	Edit | Details | Delete
Kolluri Mounav	291@gmail.com	99999	Edit | Details | Delete

© 2022 - CrudOperations_Razor - Privacy

CrudOperations_Razor Home Privacy

Edit

StudentRegistration

Stuname

K . Mounav

Email

208w1a1291@vrsec.ac.in

Salary

99

Save

Back to List

© 2022 - CrudOperations_Razor - Privacy

CrudOperations_Razor Home Privacy

VRSEC Student Registration

Create New

Stuname	Email	Salary	
Prabhuram	prabhu@gmail.com	15000	Edit | Details | Delete
Rizwanullah	208w1a1299@vrsec.ac.in	5	Edit | Details | Delete
M J N V Sai	208w1a12a0@vrsec.ac.in	95000	Edit | Details | Delete
K . Mounav	208w1a1291@vrsec.ac.in	99	Edit | Details | Delete

© 2022 - CrudOperations_Razor - Privacy

CrudOperations_Razor Home Privacy

Details

StudentRegistration

Stuname	K . Mounav
Email	208w1a1291@vrsec.ac.in
Salary	99

Edit | Back to List

© 2022 - CrudOperations_Razor - Privacy

CrudOperations_Razor Home Privacy

Delete

Are you sure you want to delete this?

StudentRegistration

Stuname	dfb
Email	fsm
Salary	435345

Delete | Back to List

© 2022 - CrudOperations_Razor - Privacy

CrudOperations_Razor Home Privacy

VRSEC Student Registration

Create New

Stuname	Email	Salary	
Prabhuram	prabhu@gmail.com	15000	Edit | Details | Delete
Rizwanullah	208w1a1299@vrsec.ac.in	5	Edit | Details | Delete
M J N V Sai	208w1a12a0@vrsec.ac.in	95000	Edit | Details | Delete
K . Mounav	208w1a1291@vrsec.ac.in	99	Edit | Details | Delete

© 2022 - CrudOperations_Razor - Privacy

References :

https://youtu.be/7960lQIYoe0

Result : Sucessfully Executed the Program.

Case Study : Stock Management System

Aim : Design a Stock Management System which manages the stocks and products and generates the Crystal Reports.

This Case study Contains The Below Components :

1. Login and Signup Component.
2. Stocks Component
3. Products Component
4. Generating The Stock Report and Products Report

This application will also interacts with DataBase connections also.

Procedure :

DataBase Part :

1. Open visual studio code and navigate to the " View Section " and then select the " Server Explorer Option "
2. Then right click on the " Data Connections " and then click on " Add Connection Option "
3. Now, select the " Microsoft SQL Server Database File (sql Client) " option
4. Then, create a Database with the name Stock management.

5. Now, create a 3 tables with the names Login, Stock, Products and their queries are below there :

Login Table Query

```
CREATE TABLE Login
(
        [UserName] VARCHAR(50) NOT NULL PRIMARY KEY,
   [Password] VARCHAR(50) NULL
 )
```

Products Table Query

```
CREATE TABLE Products
(
        [ProductCode] INT NOT NULL PRIMARY KEY,
   [ProductName] VARCHAR(150) NULL,
   [ProductStatus] BIT NULL
  )
```

Stocks Table Query

```
CREATE TABLE Stock
(
        [ProductCode] INT NOT NULL PRIMARY KEY,
   [ProductName] VARCHAR(150) NULL,
   [TransDate] DATETIME NULL,
   [Quantity] FLOAT NULL
 )
```

Login Form Part :

1. Open the New Project then select the windows form application and save it to the local machine.
2. navigate to the " View Section " and then select the " Solution Explorer Option "
3. now, right click on the project name then click on " ADD Component " and then select the windows form application and save it as "Login.cs"
4. Now, design the Login Form Just like below one .

5. now, right click on the project name then click on " ADD Component " and then select the " MDI Parent Form (Windows Forms) and save it as "StockMain.cs"

6. after adding this component you will see this is like and notepad .
7. now, remove all the items in the menu strip and the bar below the menu strip also.
8. Now, design this “StockMain.cs “ just like below one.

9. Now, we will write the C# Code for the “ StockMain.cs” and “ Login.cs”.
10. Then we will validate the Login Form details or credentials.
11. The codes for both “StockMain and login “ are there below one:

Stock Main.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace Stock_Management
{
    public partial class StockMain : Form
    {
        public StockMain()
        {
            InitializeComponent();
        }
        private void productsToolStripMenuItem_Click(object sender, EventArgs e)
        {
            // products option in Navbar
            Products pro = new Products();
            pro.MdiParent = this;
            pro.StartPosition = FormStartPosition.CenterScreen;
            pro.Show();
        }
        bool close = true;
        private void StockMain_FormClosing(object sender, FormClosingEventArgs e)
        {
```

```
            if(close)
            {
                DialogResult result = MessageBox.Show(" Are You Sure Want To Exit ",
"Exit", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
                if (result == DialogResult.Yes)
                {
                    close = false;
                    Application.Exit();
                }
                else
                {
                    e.Cancel = true;
                }
            }
        }
    }
}
```

Login Form Code

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
```

```
using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient; // add this line

namespace Stock_Management

{

    public partial class Login : Form

    {

        public Login()

        {

            InitializeComponent();

        }

        private void button1_Click(object sender, EventArgs e)

        {

            // Clear Button

            textBox1.Text = "";

            textBox2.Clear();

            textBox1.Focus();

        }

        private void button2_Click(object sender, EventArgs e)

        {

            // validating the Login Credentials

            SqlConnection con = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\SHREEE\\OneDrive\\Documents\\St
ockManagement.mdf;Integrated Security=True;Connect Timeout=30");

            con.Open();

            SqlCommand cmd = new SqlCommand("select * from Login where UserName =
@UserName and Password = @Password", con);

            cmd.Parameters.AddWithValue("@UserName", textBox1.Text);

            cmd.Parameters.AddWithValue("@Password", textBox2.Text);

            cmd.ExecuteNonQuery();
```

```
            SqlDataAdapter sda = new SqlDataAdapter(cmd);

            DataTable dt = new DataTable();

            sda.Fill(dt);

            if(dt.Rows.Count == 1)

            {

                this.Hide();

                StockMain main = new StockMain();

                main.Show();

            }

            else

            {

                MessageBox.Show("Invalid Usename and password ... !", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);

                button1_Click(sender, e);

            }

            con.Close();

            // Login Button

        }

    }

}
```

Note :

Use the Connection String of Your Own DataBase.

Connection String Part :

1. When we are writing the coding part in back of the form we mostly use the database connection string many times and this string is very large.
2. So, to reduce the database connection string we can do some minur changes in the " App.config " and the code is below there

App.Config

```
<?xml version="1.0" encoding="utf-8" ?>

<configuration>

    <startup>

        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />

    </startup>

  <connectionStrings>

    <add name ="StockConn" connectionString="Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\SHREEE\OneDrive\Documents\StockMan
agement.mdf;Integrated Security=True;Connect Timeout=30"
providerName="System.Data.SqlClient" />

  </connectionStrings>

</configuration>
```

3. After modifying the code in app.config we have to add a component.
4. now, right click on the project name then click on " ADD Component " and then select the " Class " and save it as "Connection.cs"
5. In this connection.cs file we will write the only single static function and it is invoked when we want to use the database connection string.

Connection.cs

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;
```

```
using System.Threading.Tasks;

using System.Data.SqlClient;

namespace Stock_Management

{

    public static class Connection

    {

        public static SqlConnection getConnection()

        {

            SqlConnection con = new SqlConnection();

            con.ConnectionString =
System.Configuration.ConfigurationManager.ConnectionStrings["StockConn"].ConnectionString;

            return con;

        }

    }

}
```

Products Form :

1. now, right click on the project name then click on “ ADD Component “ and then select the “ Windows Forms “ and save it as “products.cs”
2. navigate to the Form Tool Box and search for “ Error provider “ and double click on that then navigate to the Product.cs

3. When you add combo box to the status then in drop down add these 2 parameters also they are " Active & Deactive ".
4. The code for products form will below there :

Products Form Code

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;
```

```
namespace Stock_Management
{
    public partial class Products : Form
    {
        public Products()
        {
            InitializeComponent();
        }
        private void Products_Load(object sender, EventArgs e)
        {
            comboBox1.SelectedIndex = 0;
            LoadData();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            // Add button
            if(Validation())
            {
                SqlConnection con = Connection.getConnection();
                con.Open();
                bool status = false;
                if (comboBox1.SelectedIndex == 0)
                {
                    status = true;
                }
                else
                {
                    status = false;
                }
```

```
                var sqlquery = "";

                if (IfProductExists(con, textBox1.Text))

                {

                    sqlquery = "update Products set ProductName = @ProductName,
ProductStatus = @ProductStatus where ProductCode = @ProductCode";

                }

                else

                {

                    sqlquery = "insert into Products values(@ProductCode,
@ProductName, @ProductStatus)";

                }

                SqlCommand cmd = new SqlCommand(sqlquery, con);

                cmd.Parameters.AddWithValue("@ProductCode",
int.Parse(textBox1.Text));

                cmd.Parameters.AddWithValue("@ProductName", textBox2.Text);

                cmd.Parameters.AddWithValue("@ProductStatus", status);

                cmd.ExecuteNonQuery();

                con.Close();

                LoadData();

                ResetRecords(); //..

            }

        }

        private bool IfProductExists(SqlConnection con, string productCode)

        {

            SqlDataAdapter sda = new SqlDataAdapter("select 1 from Products
where ProductCode = '"+ productCode + "'", con);

            DataTable dt = new DataTable();

            sda.Fill(dt);

            if(dt.Rows.Count > 0)

            {

                return true;
```

```
            }
            else
            {
                return false;
            }
        }
        public void LoadData()
        {
            SqlConnection con = Connection.getConnection();
            con.Open();
            SqlDataAdapter sda = new SqlDataAdapter("select * from Products",
con);
            DataTable dt = new DataTable();
            sda.Fill(dt);
            dataGridView1.Rows.Clear();
            foreach (DataRow item in dt.Rows)
            {
                int n = dataGridView1.Rows.Add();
                if ((bool)item["ProductStatus"])
                {
                    dataGridView1.Rows[n].Cells[2].Value = "Active";
                }
                else
                {
                    dataGridView1.Rows[n].Cells[2].Value = "Deactive";
                }
                dataGridView1.Rows[n].Cells[0].Value =
item["ProductCode"].ToString();
                dataGridView1.Rows[n].Cells[1].Value =
item["ProductName"].ToString();
```

```
                //dataGridView1.Rows[n].Cells[0].Value =
item["ProductStatus"].ToString();

            }

        }

        private void dataGridView1_MouseDoubleClick(object sender,
MouseEventArgs e)

        {

            button1.Text = "Update"; //..

            textBox1.Text =
dataGridView1.SelectedRows[0].Cells[0].Value.ToString();

            textBox2.Text =
dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

            if(dataGridView1.SelectedRows[0].Cells[2].Value.ToString() ==
"Active")

            {

                comboBox1.SelectedIndex = 0;

            }

            else

            {

                comboBox1.SelectedIndex = 1;

            }

        }

        private void button2_Click(object sender, EventArgs e)

        {

            // Delete button

            DialogResult dialogResult = MessageBox.Show(" Are You Sure Want To
Delete ", "Message", MessageBoxButtons.YesNo);

            if(dialogResult == DialogResult.Yes)

            {

                if (Validation())

                {

                    SqlConnection con = Connection.getConnection();
```

```
                    con.Open();
                    var sqlquery = "";
                    if (IfProductExists(con, textBox1.Text))
                    {
                        sqlquery = "delete from Products where ProductCode =
@ProductCode";
                        SqlCommand cmd = new SqlCommand(sqlquery, con);
                        cmd.Parameters.AddWithValue("@ProductCode",
int.Parse(textBox1.Text));
                        cmd.ExecuteNonQuery();
                    }
                    else
                    {
                        MessageBox.Show("Record Doesn't Exists in the table
..... !");
                    }
                    con.Close();
                    LoadData();
                    ResetRecords();
                }
            }
        }
        private void ResetRecords()
        {
            textBox1.Clear();
            textBox2.Clear();
            comboBox1.SelectedIndex = -1;
            button1.Text = "Add";
            textBox1.Focus();
        }
```

```
        private void button3_Click(object sender, EventArgs e)
        {
            ResetRecords();
        }
        private bool Validation()
        {
            bool result = false;
            if(string.IsNullOrEmpty(textBox1.Text))
            {
                errorProvider1.Clear();
                errorProvider1.SetError(textBox1, " Product Code required ");
            }
            else if(string.IsNullOrEmpty(textBox2.Text))
            {
                errorProvider1.Clear();
                errorProvider1.SetError(textBox2, " Product Name Required ");
            }
            else if(comboBox1.SelectedIndex == -1)
            {
                errorProvider1.Clear();
                errorProvider1.SetError(comboBox1, " Select the status ");
            }
            else
            {
                return true;
            }
            return result;
        }
    }
```

```
}
```

Note :

Use the Connection String of Your Own DataBase.

Stock Form :

1. now, right click on the project name then click on " ADD Component " and then select the " Windows Forms " and save it as "stock.cs"
2. navigate to the Form Tool Box and search for " Error provider " and double click on that then navigate to the stock.cs

3. When you add combo box to the status then in drop down add these 2 parameters also they are " Active & Deactive ".
4. The code for Stock form will below there :

Stock.cs Code

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
namespace Stock_Management
{
    public partial class Stock : Form
    {
        public Stock()
        {
            InitializeComponent();
        }
        private void Stock_Load(object sender, EventArgs e)
        {
            this.ActiveControl = dateTimePicker1;
            comboBox1.SelectedIndex = 0;
```

```
        LoadData();
        Search();
    }
    private void dateTimePicker1_KeyDown(object sender, KeyEventArgs e)
    {
        if(e.KeyCode == Keys.Enter)
        {
            textBox1.Focus();
        }
    }
    private void textBox1_KeyDown(object sender, KeyEventArgs e)
    {
        if(e.KeyCode == Keys.Enter)
        {
            if(dgview.Rows.Count > 0)
            {
                textBox1.Text = dgview.SelectedRows[0].Cells[0].Value.ToString();
                textBox2.Text = dgview.SelectedRows[0].Cells[1].Value.ToString();
                this.dgview.Visible = false;
                textBox3.Focus();
            }
            else
            {
                this.dgview.Visible = false;
            }
```

```
        }
    }
    private void textBox2_KeyDown(object sender, KeyEventArgs e)
    {
        if(e.KeyCode == Keys.Enter)
        {
            if(textBox2.Text.Length > 0)
            {
                textBox3.Focus();
            }
            else
            {
                textBox2.Focus();
            }
        }
    }
    bool change = true;
    private void proCode_MouseDoubleClick(object sender, MouseEventArgs e)
    {
        if (change)
        {
            change = false;
            textBox1.Text = dgview.SelectedRows[0].Cells[0].Value.ToString();
            textBox2.Text = dgview.SelectedRows[0].Cells[1].Value.ToString();
            this.dgview.Visible = false;
```

```
            textBox3.Focus();
            change = true;
        }
    }
    private void textBox3_KeyDown(object sender, KeyEventArgs e)
    {
        if(e.KeyCode == Keys.Enter)
        {
            if(textBox3.Text.Length > 0)
            {
                comboBox1.Focus();
            }
            else
            {
                textBox3.Focus();
            }
        }
    }
    private void comboBox1_KeyDown(object sender, KeyEventArgs e)
    {
        if(e.KeyCode == Keys.Enter)
        {
            if(comboBox1.SelectedIndex != -1)
            {
                button1.Focus();
```

```
        }
        else
        {
            comboBox1.Focus();
        }
    }
}
private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    if(!char.IsNumber(e.KeyChar) & (Keys)e.KeyChar != Keys.Back & e.KeyChar != '.')
    {
        e.Handled = true;
    }
}
private void textBox3_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!char.IsNumber(e.KeyChar) & (Keys)e.KeyChar != Keys.Back & e.KeyChar != '.')
    {
        e.Handled = true;
    }
}
private void ResetRecords()
{
    dateTimePicker1.Value = DateTime.Now;
    textBox1.Clear();
```

```
        textBox2.Clear();
        textBox3.Clear();
        comboBox1.SelectedIndex = -1;
        button1.Text = "Add";
        dateTimePicker1.Focus();
    }
    private void button3_Click(object sender, EventArgs e)
    {
        ResetRecords();
    }
    private bool Validation()
    {
        bool result = false;


        if(string.IsNullOrEmpty(textBox1.Text))
        {
            errorProvider1.Clear();
            errorProvider1.SetError(textBox1, " Product Code Required ");
        }
        else if(string.IsNullOrEmpty(textBox2.Text))
        {
            errorProvider1.Clear();
            errorProvider1.SetError(textBox2, " Product Name Required ");
        }
        else if(string.IsNullOrEmpty(textBox3.Text))
```

```
        {
            errorProvider1.Clear();
            errorProvider1.SetError(textBox3, " Quantity Required ");
        }
        else if(comboBox1.SelectedIndex == -1)
        {
            errorProvider1.Clear();
            errorProvider1.SetError(comboBox1, " Select Status ");
        }
        else
        {
            errorProvider1.Clear();
            result = true;
        }
        return result;
    }
    private bool IfProductExists(SqlConnection con, string productCode)
    {
        SqlDataAdapter sda = new SqlDataAdapter("select 1 from Stock where ProductCode = '" +
productCode + "'", con);
        DataTable dt = new DataTable();
        sda.Fill(dt);
        if (dt.Rows.Count > 0)
        {
            return true;
```

```
        }
        else
        {
            return false;
        }
    }
    private void button1_Click(object sender, EventArgs e)
    {
        // Add Button
        if(Validation())
        {
            SqlConnection con = Connection.getConnection();
            con.Open();
            bool status = false;
            if(comboBox1.SelectedIndex == 0)
            {
                status = true;
            }
            else
            {
                status = false;
            }
            var sqlquery = "";
            if (IfProductExists(con, textBox1.Text))
            {
```

```
                sqlquery = "update Stock set ProductName = @ProductName, ProductStatus =
@ProductStatus, Quantity = @Quantity where ProductCode = @ProductCode";

            }

            else

            {

                sqlquery = "insert into Stock values(@ProductCode, @ProductName, @TransDate,
@Quantity, @ProductStatus)";

            }

            SqlCommand cmd = new SqlCommand(sqlquery, con);

            cmd.Parameters.AddWithValue("@ProductCode", int.Parse(textBox1.Text));

            cmd.Parameters.AddWithValue("@ProductName", textBox2.Text);

            cmd.Parameters.AddWithValue("@TransDate",
dateTimePicker1.Value.ToString("MM/dd/yyyy"));

            cmd.Parameters.AddWithValue("@Quantity", textBox3.Text);

            cmd.Parameters.AddWithValue("@ProductStatus", status);

            cmd.ExecuteNonQuery();

            con.Close();

            MessageBox.Show(" Record Saved Successfully ");

            LoadData();



            ResetRecords(); //..

        }

    }

    public void LoadData()

    {

        SqlConnection con = Connection.getConnection();
```

```
        SqlDataAdapter sda = new SqlDataAdapter("select * from Stock", con);

        DataTable dt = new DataTable();

        sda.Fill(dt);

        dataGridView1.Rows.Clear();

        foreach(DataRow item in dt.Rows)

        {

           int n = dataGridView1.Rows.Add();

           dataGridView1.Rows[n].Cells["dgSno"].Value = n + 1;

           dataGridView1.Rows[n].Cells["dgProCode"].Value = item["ProductCode"].ToString();

           dataGridView1.Rows[n].Cells["dgProName"].Value = item["ProductName"].ToString();

           dataGridView1.Rows[n].Cells["dgQuantity"].Value = float.Parse(item["Quantity"].ToString());

           dataGridView1.Rows[n].Cells["dgDate"].Value =
Convert.ToDateTime(item["TransDate"].ToString()).ToString("dd/MM/yyyy");

           if ((bool)item["ProductStatus"])

           {

              dataGridView1.Rows[n].Cells["dgStatus"].Value = "Active";

           }

           else

           {

              dataGridView1.Rows[n].Cells["dgStatus"].Value = "Deactive";

           }

        }

        if(dataGridView1.Rows.Count > 0)

        {

           label8.Text = dataGridView1.Rows.Count.ToString();
```

```
            float totQty = 0;
            for(int i = 0; i < dataGridView1.Rows.Count; ++i)
            {
                totQty += float.Parse(dataGridView1.Rows[i].Cells["dgQuantity"].Value.ToString());
                label9.Text = totQty.ToString();
            }
        }
        else
        {
            label8.Text = "0";
            label9.Text = "0";
        }
    }
    private void dataGridView1_MouseDoubleClick(object sender, MouseEventArgs e)
    {
        button1.Text = "Update";
        textBox1.Text = dataGridView1.SelectedRows[0].Cells["dgProCode"].Value.ToString();
        textBox2.Text = dataGridView1.SelectedRows[0].Cells["dgProName"].Value.ToString();
        textBox3.Text = dataGridView1.SelectedRows[0].Cells["dgQuantity"].Value.ToString();
        dateTimePicker1.Text =
DateTime.Parse(dataGridView1.SelectedRows[0].Cells["dgDate"].Value.ToString()).ToString("dd/MM/y
yyy");
        if (dataGridView1.SelectedRows[0].Cells["dgStatus"].Value.ToString() == "Active")
        {
            comboBox1.SelectedIndex = 0;
```

```
        }
        else
        {
            comboBox1.SelectedIndex = 1;
        }
    }
    private void button2_Click(object sender, EventArgs e)
    {
        // Delete Button
        DialogResult dialogResult = MessageBox.Show(" Are You Sure Want To Delete ", "Message",
MessageBoxButtons.YesNo);
        if (dialogResult == DialogResult.Yes)
        {
            if (Validation())
            {
                SqlConnection con = Connection.getConnection();
                con.Open();
                var sqlquery = "";
                if (IfProductExists(con, textBox1.Text))
                {
                    sqlquery = "delete from Stock where ProductCode = @ProductCode";
                    SqlCommand cmd = new SqlCommand(sqlquery, con);
                    cmd.Parameters.AddWithValue("@ProductCode", int.Parse(textBox1.Text));
                    cmd.ExecuteNonQuery();
                }
```

```
            else
            {
                MessageBox.Show("Record Doesn't Exists in the table ..... !");
            }
            con.Close();
            LoadData();
            ResetRecords();
        }
    }
}
private void textBox1_TextChanged(object sender, EventArgs e)
{
    // product Code textbox
    if(textBox1.Text.Length > 0)
    {
        this.dgview.Visible = true;
        dgview.BringToFront();
        Search(150, 105, 430, 200, "Pro Code, Pro Name", "100");
        // To do : mouse Double Click Event
        this.dgview.MouseDoubleClick += new
System.Windows.Forms.MouseEventHandler(this.proCode_MouseDoubleClick);
        SqlConnection con = Connection.getConnection();
        con.Open();
```

```
            SqlCommand cmd = new SqlCommand("select Top(10) ProductCode, ProductName from
Products where ProductCode like @ProductCode", con);

            cmd.Parameters.AddWithValue("@ProductCode", int.Parse(textBox1.Text));

            SqlDataAdapter sda = new SqlDataAdapter(cmd);

            DataTable dt = new DataTable();

            sda.Fill(dt);

            dgview.Rows.Clear();

            foreach(DataRow row in dt.Rows)

            {

               int n = dgview.Rows.Add();

               dgview.Rows[n].Cells[0].Value = row["ProductCode"].ToString();

               dgview.Rows[n].Cells[1].Value = row["ProductName"].ToString();

            }

         }

         else

         {

            dgview.Visible = false;

         }

      }

      private DataGridView dgview;

      private DataGridViewTextBoxColumn dgviewcol1;

      private DataGridViewTextBoxColumn dgviewcol2;

      void Search()

      {

         dgview = new DataGridView();
```

```
        dgviewcol1 = new DataGridViewTextBoxColumn();

        dgviewcol2 = new DataGridViewTextBoxColumn();

        this.dgview.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;

        this.dgview.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {
this.dgviewcol1, this.dgviewcol2 });

        this.dgview.Name = "dgview";

        dgview.Visible = false;

        this.dgviewcol2.AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

        this.dgviewcol1.Visible = false;

        this.dgviewcol2.Visible = false;

        this.dgview.AllowUserToAddRows = false;

        this.dgview.RowHeadersVisible = false;

        this.dgview.SelectionMode =
System.Windows.Forms.DataGridViewSelectionMode.FullRowSelect;

        //this.dgview.KeyDown += new
System.Windows.Forms.KeyEventHandler(this.dgview_KeyDown);

        this.Controls.Add(dgview);

        this.dgview.ReadOnly = true;

        dgview.BringToFront();

    }

    //Two Column

    void Search(int LX, int LY, int DW, int DH, string ColName, String ColSize)

    {

        this.dgview.Location = new System.Drawing.Point(LX, LY);

        this.dgview.Size = new System.Drawing.Size(DW, DH);
```

```
        string[] ClSize = ColSize.Split(',');
        //Size
        for(int i = 0; i < ClSize.Length; i++)
        {
            if(int.Parse(ClSize[i]) != 0)
            {
                dgview.Columns[i].Width = int.Parse(ClSize[i]);
            }
            else
            {
                dgview.Columns[i].AutoSizeMode =
System.Windows.Forms.DataGridViewAutoSizeColumnMode.Fill;
            }
        }
        //Name
        string[] ClName = ColName.Split(',');
        for (int i = 0; i < ClName.Length; i++)
        {
            this.dgview.Columns[i].HeaderText = ClName[i];
            this.dgview.Columns[i].Visible = true;
        }
    }
  }
}
```

Installation Of The Crystal Reporting Services Tool

1. Before we go for the generating the reports for the products and stock windows form.
2. Make sure that your visual studio version is 2022.
3. First we have to install the crystal reporting services tool in your systems.
4. Download the crystal reporting services from the below link https://drive.google.com/file/d/1AEuZ2AM-5B1daK0BAQEhRQCNqHbe-dim/view?usp=sharing

5. After downloading the file then unzip the rar file you will see there are 3 files in that folder.

6. First run the " Second File " and give all permissons and access to the tool.
7. Next run the " Last File " and give all permissons and access to the tool.

Sample Installation Pic :

Dataset Creation Part

1. Open the solution explorer and right click on project name and create a 3 new folders with Names “ Datas “, “ ReportForm “ “ Reports “ just like below one.

2. Now right click on Project name and choose “ Add item option” And then select the Dataset option.

3. After creating it “ Double click on the DataSet1.xsd “ and just drag the Products and Stock tables from the server explorer into the DataSet1.xsd

Product Report Part

1. Right click on the reports folder and then add an item crystal reporting and save as Product.rpt

2. Select the Blank report option and click on ok.

3. Now right click on the section 1 and select the page setup
   Change the Letter format to the A4 sheet format.

4. To write on the report , to print the date , row number in any section you just right click on that section you will get the options releated to that.
   For example you want textfield you can select or you date option then go to the special field.

5. The final report template will be like below one.

6. You can design the section 1, section 2, section 4, section 5 in the report in your way but the section 3 is connected to the dataset and database.
7. On left side panel click on the Field Explorer tab and then right click on the Database option then choose " database expert " Option.
8. Now send the " Products table " from the available data sources to the selected tables list then click on ok.

9. Now drag the Product code , product name fields from the database fields to the “ Details Section “.

10. In details section you can record number box just like below one.

11. With this the Product report template is completed.

Product Report Windows Form Part

1. Right click on the reportform folder and add a windows form and save it as a name ProductReport.cs

2. Now select the " crystal report viwer tool " from the tool box and drag into the windows form.

3. Double click on the top of the windows form then it will open the Form load section code.

ProductReport.cs File Code

```
using CrystalDecisions.CrystalReports.Engine;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using Stock_Management.Datas;
namespace Stock_Management.ReportForm
{
    public partial class ProductsReport : Form
    {
        ReportDocument cryrpt = new ReportDocument();
        public ProductsReport()
        {
            InitializeComponent();
        }
        private void ProductsReport_Load(object sender, EventArgs e)
        {
            // Load Form
```

```
            cryrpt.Load(@"E:\venkat
sai\Dot_Net\Csharp\Stock_Management\Stock_Management\Reports\Product.rpt");

            SqlConnection con = Connection.getConnection();

            con.Open();

            DataSet dst = new DataSet();

            SqlDataAdapter sda = new SqlDataAdapter("Select * From Products", con);

            DataTable dt = new DataTable();

            sda.Fill(dt);

            cryrpt.SetDataSource(dt);

            crystalReportViewer1.ReportSource = cryrpt;

        }

    }

}
```

Stock Report Part

1. Right click on the reports folder and then add an item crystal reporting and save as Stock.rpt

2. Select the blank report as an option and click on ok.

3. Now right click on the section 1 and select the page setup Change the Letter format to the A4 sheet format.

4. Now design the stock report template in your way and in header section you have print the from date to the to date date.

5. We can achive that by using the parameter fields in the field explorer tab.

6. Click on field tools tab on the left side panel there right click on the parameter fields and click on NEW.
7. You have to 2 parameters fields one is FromDate field and ToDate Field.

8. After adding both parameters into the parameter fields. Just drag the FromDate and ToDate fields from the parameter fields to the Section 1 (Report header).
9. Now right click on the database fields and choose the database expert option.
10. Now send the " Products table " from the available data sources to the selected tables list then click on ok.

11. Now drag the Product code , product name, quantity fields from the database fields to the “ Details Section “.

12. In details section you can record number box just like below one.

13. With these the Stock report template is completed.

Stock Report Windows Form Part

1. Right click on the reportform folder and add a windows form and save it as a name StockReport.cs

2. Now select the " crystal report viwer tool " from the tool box and drag into the windows form and design the form just like below one..

3. Double click on the Show report button then it will open the Button section code.

StockReport.cs Form Code

```
using CrystalDecisions.CrystalReports.Engine;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
namespace Stock_Management.ReportForm
{
    public partial class StockReport : Form
    {
        ReportDocument crystal = new ReportDocument();
        public StockReport()
        {
            InitializeComponent();
        }
        private void StockReport_Load(object sender, EventArgs e)
        {
            // Form load Section
        }
```

```
        private void button1_Click(object sender, EventArgs e)

        {

            // Show Report

            crystal.Load(@"E:\venkat
sai\Dot_Net\Csharp\Stock_Management\Stock_Management\Reports\Stock.rpt");

            SqlConnection con = Connection.getConnection();

            con.Open();

            DataSet dst = new DataSet();

            SqlDataAdapter sda = new SqlDataAdapter("Select * From Stock where Cast(
TransDate as Date) between '" + dateTimePicker1.Value.ToString("MM/dd/yyyy") + "' and '" +
dateTimePicker2.Value.ToString("MM/dd/yyyy") + "'", con);

            sda.Fill(dst, "Stock");

            crystal.SetDataSource(dst);

            crystal.SetParameterValue("@FromDate",
dateTimePicker1.Value.ToString("dd/MM/yyyy"));

            crystal.SetParameterValue("@ToDate",
dateTimePicker2.Value.ToString("dd/MM/yyyy"));

            crystalReportViewer1.ReportSource = crystal;

        }

    }

}
```

Update the StockMain Windows Form Part

Updated StockMain.cs Form Code

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;
```

```
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace Stock_Management
{
    public partial class StockMain : Form
    {
        public StockMain()
        {
            InitializeComponent();
        }
        private void productsToolStripMenuItem_Click(object sender, EventArgs e)
        {
            // products option in Navbar
            Products pro = new Products();
            pro.MdiParent = this;
            pro.StartPosition = FormStartPosition.CenterScreen;
            pro.Show();
        }
        bool close = true;
        private void StockMain_FormClosing(object sender, FormClosingEventArgs e)
        {
            if(close)
            {
```

```
            DialogResult result = MessageBox.Show(" Are You Sure Want To Exit ", "Exit",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);

            if (result == DialogResult.Yes)

            {

                close = false;

                Application.Exit();

            }

            else

            {

                e.Cancel = true;

            }

        }

    }

    private void stockToolStripMenuItem_Click(object sender, EventArgs e)

    {

        // Stock

        Stock stk = new Stock();

        stk.MdiParent = this;

        stk.StartPosition = FormStartPosition.CenterScreen;

        stk.Show();

    }

    private void productListToolStripMenuItem_Click(object sender, EventArgs e)

    {

        // Products Report

        ReportForm.ProductsReport prod = new ReportForm.ProductsReport();
```

```
            prod.MdiParent = this;
            prod.StartPosition = FormStartPosition.CenterScreen;
            prod.Show();
        }
        private void stockListToolStripMenuItem_Click(object sender, EventArgs e)
        {
            // Stock Report
            ReportForm.StockReport prod = new ReportForm.StockReport();
            prod.MdiParent = this;
            prod.StartPosition = FormStartPosition.CenterScreen;
            prod.Show();
        }
    }
}
```

Final Outputs :

Invalid Login :

Valid Login :

Products Form :

Button : Add

Button : Update

Button : Delete

Stock Form :

Button : Add

Button : Update

S.No	Product Code	Product Name	Quantity	Date	Status
1	12	Oil 34L	2	05-11-2022	Active
2	56	Cricket Bat	10	15-11-2022	Active
3	66	Charger	3	05-11-2022	Active
4	67	oil 1L	45	05-11-2022	Deactive
5	68	Asus laptop	110	05-11-2022	Active
6	83	Iphone 14 pro	2	13-11-2022	Active
7	234	Dolo 64	20	10-11-2022	Deactive
8	420	Room Spray	520	10-11-2022	Active
9	569	lenovo I7	58	05-11-2022	Deactive
10	898	laptop	2	05-11-2022	Active

S.No	Product Code	Product Name	Quantity	Date	Status
1	12	Oil 34L	2	05-11-2022	Active
2	56	Cricket Bat	10	15-11-2022	Deactive
3	66	Charger	3	05-11-2022	Active
4	67	oil 1L	45	05-11-2022	Deactive
5	68	Asus laptop	110	05-11-2022	Active
6	83	Iphone 14 pro	2	13-11-2022	Active
7	234	Dolo 64	20	10-11-2022	Deactive
8	420	Room Spray	520	10-11-2022	Active
9	569	lenovo I7	58	05-11-2022	Deactive
10	898	laptop	2	05-11-2022	Active

Button : Delete

S.No	Product Code	Product Name	Quantity	Date	Status
1	12	Oil 34L	2	05-11-2022	Active
2	56	Cricket Bat	10	15-11-2022	Deactive
3	66	Charger	3	05-11-2022	Active
4	67	oil 1L	45	05-11-2022	Deactive
5	68	Asus laptop	110	05-11-2022	Active
6	234	Dolo 64	20	10-11-2022	Deactive
7	420	Room Spray	520	10-11-2022	Active
8	569	lenovo I7	58	05-11-2022	Deactive
9	898	laptop	2	05-11-2022	Active

Product List Report :

Stock List Report :

Result : Sucessfully Executed the Application.

References :

https://youtube.com/playlist?list=PL7FAIXDZVr6bNaAIvmNd9q65mlAIu7XI6

https://youtu.be/lVAI9M_yIH0

https://youtu.be/JA52XDHJk9o