Ganpat University

॥ विद्यया समाजोत्कर्षः ॥

A.M. Patel Institute of Computer Studies

WEB TECHNOLOGY USING .NET (P12A2WTD)

MCA SEM-2

VIDHEY H. DETROJA
(23034211007)

No.	Practical Name	Date	Sign	Page
1.	Implement C# Program to Calculates the Distance Travelled by Reading Speed and Time.	14/02/2024		
2.	Implement C# Program to calculate perimeter of circle and rectangle.	16/02/2024		
3.	Implement C# Program to check whether the entered number is even or odd.	19/02/2024		
4.	Implement C# Program to check whether the entered number is an Armstrong number or not.	20/02/2024		
5.	Implement C# Program to Reads a Grade & Display the Equivalent Description.(Switch Case)	22/02/2024		
6.	Implement C# Program to Displays the ATM Transaction. Here The types of ATM transaction are 1) Balance checking 2) Cash withdrawal 3) Cash deposition. You can opt any of the above transaction according to your need of transaction. (Switch Case)	23/02/2024		
7.	Implement C# Program Prints a Binary Triangle.(Using For Loop) Problem Solution: Binary Triangle is a Triangle formed with 1's and 0's.Number of rows in the binary triangle is obtained from the user.	27/02/2024		
8.	Implement c# program to demonstrate the use of Continue and goto Statement.	29/02/2024		
9.	Implement C# Program Gets a Number and Display the Number with its Reverse.(Using While Loop)	01/03/2024		
10.	Implement C# Program to reverse an array. (Using for each Loop)	04/03/2024		
11.	Implement C# Program to search an element in an array.	05/03/2024		
12.	Implement C# Program to Demonstrates Jagged Arrays.	07/03/2024		
13.	Implement c# program to list all substrings in a	08/03/2024		

	given string.			
14.	Implement c# program to demonstrate the use of following String Methods. ToUpper(), ToLOwer(), Concat(), IndexOf(), Substring()	12/03/2024		
15.	Implement c# program to demonstrate the use of Access Modifiers.	21/03/2024		
16.	Implement c# program to demonstrate the use of Constructor and Destructor.	22/03/2024		
17.	Implement c# program to demonstrate the use of Inheritance (Single Inheritance, Multi-Level Inheritance)	29/03/2024		
18.	Implement c# program to demonstrate the use of Polymorphism (Method Overloading and Method Overriding)	04/04/2024		
19.	Implement c# program to demonstrate the use of the following methods of ArrayList Class. Add(), Insert(), Remove(), RemoveAt(), Sort(), Reverse(), Contains(), IndexOf()	15/04/2024		
20.	Implement c# program to demonstrate the use of the following methods of Hashtable Class. Add(), Remove(), ContainsKey(), ContainsValue()	16/04/2024		
21.	Implement c# program to demonstrate the use of Exception Handling. (try, catch, finally)	16/04/2024		
22.	Implement c# program to demonstrate the use of LINQ Query Syntax and LINQ Method Syntax.	18/04/2024		
23.	Implement c# program to demonstrate the use of LINQ Query Operators (select, where, OrderBy, Count, Max, Min, Sum, All, Any, Contains, etc.)	19/04/2024		
24.	Develop ASP.NET Core MVC web application to demonstrate the use of ViewData, Viewbag and TempData.	22/04/2024		
25.	Develop ASP.NET Core MVC web application to Create a Layout View in ASP.NET Core MVC.	24/04/2024		
26.	Develop ASP.NET Core MVC web application to demonstrate the use of Razor Syntax (If …else, Switch, For Loop, For each Loop, While Loop)	25/04/2024		

27.	**Create ASP.NET Core MVC registration page using Razor Syntax and Html Helper methods to register new student.** **First name (TextBox)** **Last name or initial (TextBox)** **Gender (Radio Button)** **Age (Take numeric input only)** **Existing Qualification(Dropdown List)** **Email address (Regular Expression)** **Choose a password (min. 8 characters)**	**26/04/2024**		
28.	**Develop ASP.NET core MVC web application to perform CRUD operation by using entity framework with validation on Book model.** **Book (BookId, BookTitle, AuthorName, Publication and Price)**	**29/04/2024**		
29.	**Develop ASP.NET Core MVC application to perform CRUD operation by using entity framework with validation on Student, Course and Enrollment models.**	**30/04/2024**		

1.	**Implement C# Program to Calculates the Distance Travelled by Reading Speed and Time.**
Code	

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading.Tasks;
namespace mehtods
{
    class Program
    {
        static void Main()
        {
            double speed, time, distance;
            Console.WriteLine("Develop by Vidhey H. Detroja");
            Console.WriteLine("Enroll no. - 23034211007");
            Console.Write("Enter speed in km/h: ");
            speed = Convert.ToDouble(Console.ReadLine());
            Console.Write("Enter time in hours: ");
            time = Convert.ToDouble(Console.ReadLine());
            distance = speed * time;
            Console.WriteLine("Distance travelled: {0} kilometers", distance);
        }
    }
}
```

Output

```
Develop by Vidhey H. Detroja
Enroll no. - 23034211007
Enter speed in km/h: 219
Enter time in hours: 4
Distance travelled: 876 kilometers
```

2.	**Implement C# Program to calculate perimeter of circle and rectangle.**
Code	

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading.Tasks;
namespace mehtods
{
   class Program
   {
      static void Main()
      {
         Console.WriteLine("Develop by Vidhey H. Detroja");
         Console.WriteLine("Enroll no.- 2303421007");
         Console.Write("Enter the radius of the circle: ");
         double radius = Convert.ToDouble(Console.ReadLine());
         double circlePerimeter = 2 * 3.14 * radius;
         Console.WriteLine("Perimeter of the circle: {0}", circlePerimeter);
         Console.Write("\nEnter the length of the rectangle: ");
         double length = Convert.ToDouble(Console.ReadLine());
         Console.Write("Enter the width of the rectangle: ");
         double width = Convert.ToDouble(Console.ReadLine());
         double rectanglePerimeter = 2 * (length + width);
         Console.WriteLine("Perimeter of the rectangle: {0}", rectanglePerimeter);
      }
   }
}
```

Output

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
Enter the radius of the circle: 11
Perimeter of the circle: 69.08

Enter the length of the rectangle: 19
Enter the width of the rectangle: 21
Perimeter of the rectangle: 80
```

3.	**Implement C# Program to check whether the entered number is even or odd.**
Code	

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading.Tasks;
namespace mehtods
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("Develop by Vidhey H. Detroja");
            Console.WriteLine("Enroll no.- 2303421007");
            Console.Write("Enter a number: ");
            int number = Convert.ToInt32(Console.ReadLine());
            if (number % 2 == 0)
            {
                Console.WriteLine("{0} is an even number.", number);
            }
            else{
                Console.WriteLine("{0} is an odd number.", number);
            }
        }
    }
}
```

Output

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
Enter a number: 19
19 is an odd number.
```

4.	**Implement C# Program to check whether the entered number is an Armstrong number or not.**
Code	

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading.Tasks;
namespace mehtods
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("Develop by Vidhey H. Detroja");
            Console.WriteLine("Enroll no.- 2303421007");
            Console.Write("Enter a number: ");
            int number = Convert.ToInt32(Console.ReadLine());
            int numDigits = 0;
            int originalNumber = number;
            int sum = 0;
            while (number != 0)
            {
                numDigits++;
                number /= 10;
            }
            number = originalNumber;
            while (number != 0)
            {
                int digit = number % 10;
                sum += (int)Math.Pow(digit, numDigits);
```

```
            number /= 10;
        }
        if (sum == originalNumber)
        {
            Console.WriteLine("{0} is an Armstrong number.", originalNumber);
        }
        else
        {
            Console.WriteLine("{0} is not an Armstrong number.", originalNumber);
        }
    }
}
}
```

Output

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
Enter a number: 1119
1119 is not an Armstrong number.
```

5.	**Implement C# Program to Reads a Grade & Display the Equivalent Description.(Switch Case)**
Code	

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Develop by Vidhey H. Detroja");
        Console.WriteLine("Enroll no.- 2303421007");
        Console.WriteLine("Enter your grade (A, B, C, D, or F):");
        char grade = Convert.ToChar(Console.ReadLine().ToUpper());
        string description;
        switch (grade)
        {
            case 'A':
                description = "Excellent";
                break;
            case 'B':
                description = "Good";
                break;
            case 'C':
                description = "Average";
                break;
            case 'D':
                description = "Below Average";
                break;
            case 'F':
                description = "Fail";
                break;
            default:
                description = "Invalid grade";
```

	break; } Console.WriteLine("Grade Description: " + description); } }
Output	Develop by Vidhey H. Detroja Enroll no.- 2303421007 Enter your grade (A, B, C, D, or F): A Grade Description: Excellent

6.	**Implement C# Program to Displays the ATM Transaction.** **Here The types of ATM transaction are** **1) Balance checking** **2) Cash withdrawal** **3) Cash deposition.** **You can opt any of the above transaction according to your need of transaction. (Switch Case)**

Code	using System;
	
	class Program
	
	{
	
	static void Main(string[] args)
	
	{
	
	Console.WriteLine("Develop by Vidhey H. Detroja");
	
	Console.WriteLine("Enroll no.- 2303421007");
	
	Console.WriteLine("Welcome to the ATM!");
	
	Console.WriteLine("Choose your transaction type:");
	
	Console.WriteLine("1) Balance checking");
	
	Console.WriteLine("2) Cash withdrawal");
	
	Console.WriteLine("3) Cash deposition");
	
	int choice = Convert.ToInt32(Console.ReadLine());
	
	switch (choice)
	
	{
	
	case 1:
	
	Console.WriteLine("You selected: Balance checking");
	
	break;
	
	case 2:
	
	Console.WriteLine("You selected: Cash withdrawal");
	
	break;
	
	case 3:
	
	Console.WriteLine("You selected: Cash deposition");

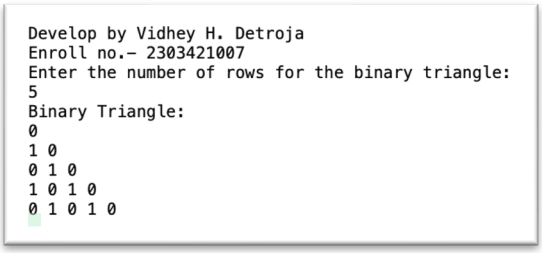
```
            break;
        default:
            Console.WriteLine("Invalid choice. Please select a valid transaction type.");
            break;
    }
  }
}
```

Output

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
Welcome to the ATM!
Choose your transaction type:
1) Balance checking
2) Cash withdrawal
3) Cash deposition
1
You selected: Balance checking
```

7.	**Implement C# Program Prints a Binary Triangle.(Using For Loop)** **Problem Solution: Binary Triangle is a Triangle formed with 1's and 0's.Number of rows in the binary triangle is obtained from the user.**

Code	using System;
	
	class Program
	
	{
	
	static void Main(string[] args)
	
	{
	
	Console.WriteLine("Develop by Vidhey H. Detroja");
	
	Console.WriteLine("Enroll no.- 2303421007");
	
	Console.WriteLine("Enter the number of rows for the binary triangle:");
	
	int numRows = Convert.ToInt32(Console.ReadLine());
	
	Console.WriteLine("Binary Triangle:");
	
	for (int i = 1; i <= numRows; i++){
	
	for (int j = 1; j <= i; j++){
	
	if ((i + j) % 2 == 0){
	
	Console.Write("0 ");
	
	}else{
	
	Console.Write("1 ");
	
	}
	
	}
	
	Console.WriteLine(); // Move to the next line after printing each row
	
	}
	
	}
	
	}

Output	 Develop by Vidhey H. Detroja Enroll no.- 2303421007 Enter the number of rows for the binary triangle: 5 Binary Triangle: 0 1 0 0 1 0 1 0 1 0 0 1 0 1 0

8.	**Implement c# program to demonstrate the use of Continue and goto Statement.**
Code	

```
using System;

class Program

{

   static void Main(string[] args)

   {

      Console.WriteLine("Develop by Vidhey H. Detroja");

      Console.WriteLine("Enroll no.- 2303421007");

      Console.WriteLine("Demonstrating the use of continue statement:");

      for (int i = 1; i <= 5; i++){

         if (i == 3{

            continue;

         }

         Console.WriteLine(i);

      }

      Console.WriteLine("\nDemonstrating the use of goto statement:");

      int count = 0;

      startLoop:

      count++;

      if (count <= 5){

         Console.WriteLine(count);

         goto startLoop;

      }

   }

}
```

Output

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
Demonstrating the use of continue statement:
1
2
4
5

Demonstrating the use of goto statement:
1
2
3
4
5
```

9.	**Implement C# Program Gets a Number and Display the Number with its Reverse.(Using While Loop)**
Code	

```
using System;

class Program

{

   static void Main(string[] args)

   {

      Console.WriteLine("Develop by Vidhey H. Detroja");

      Console.WriteLine("Enroll no.- 2303421007");

      Console.WriteLine("Enter a number:");

      int number = Convert.ToInt32(Console.ReadLine());

      int originalNumber = number;

      int reverse = 0;

      while (number > 0)

      {

         int digit = number % 10;

         reverse = reverse * 10 + digit;

         number /= 10;

      }

      Console.WriteLine("Original number: " + originalNumber);

      Console.WriteLine("Reverse number: " + reverse);

   }

}
```

Output

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
Enter a number:
1911
Original number: 1911
Reverse number: 1191
```

10.	**Implement C# Program to reverse an array. (Using for each Loop)**
Code	

```
class Program
{
    static void Main(string[] args)
    {
        int[] array = { 1, 2, 3, 4, 5 };
        Console.WriteLine("Develop by Vidhey H. Detroja");
        Console.WriteLine("Enroll no.- 2303421007");
        Console.WriteLine("Original array:");
        PrintArray(array);
        ReverseArray(array);
        Console.WriteLine("\nReversed array:");
        PrintArray(array);
    }
    static void ReverseArray(int[] arr)
    {
        int[] reversedArray = new int[arr.Length];
        int index = 0;
        foreach (int element in arr)
        {
            reversedArray[arr.Length - 1 - index] = element;
            index++;
        }
        for (int i = 0; i < arr.Length; i++)
        {
            arr[i] = reversedArray[i];
        }
    }
    static void PrintArray(int[] arr)
    {
```

```
        foreach (int element in arr)
        {
            Console.Write(element + " ");
        }
        Console.WriteLine();
    }
}
```

Output

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
Original array:
1 2 3 4 5

Reversed array:
5 4 3 2 1
```

11. **Implement C# Program to search an element in an array.**

Code

```
using System;

class Program

{

    static void Main(string[] args)

    {

        Console.WriteLine("Develop by Vidhey H. Detroja");

        Console.WriteLine("Enroll no.- 2303421007");

        int[] array = { 2, 4, 6, 8, 10, 12, 14, 16 };

        int elementToSearch = 10;

        int index = SearchElement(array, elementToSearch);

        if (index != -1){

            Console.WriteLine("Element " + elementToSearch + " found at index " + index);

        } else{

            Console.WriteLine("Element " + elementToSearch + " not found in the array.");

        }

    }

    static int SearchElement(int[] arr, int element){

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

            if (arr[i] == element){

                return i; // Return the index where the element is found

            }

        }

        return -1; // Return -1 if the element is not found in the array

    }

}
```

Output

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
Element 10 found at index 4
```

12.	**Implement C# Program to Demonstrates Jagged Arrays.**
Code	

```
using System;
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Develop by Vidhey H. Detroja");
        Console.WriteLine("Enroll no.- 2303421007");
        int[][] jaggedArray = new int[3][];
        jaggedArray[0] = new int[] { 1, 2, 3 };
        jaggedArray[1] = new int[] { 4, 5 };
        jaggedArray[2] = new int[] { 6, 7, 8, 9 };
        Console.WriteLine("Jagged Array:");
        for (int i = 0; i < jaggedArray.Length; i++)
        {
            Console.Write("Row " + i + ": ");
            for (int j = 0; j < jaggedArray[i].Length; j++)
            {
                Console.Write(jaggedArray[i][j] + " ");
            }
            Console.WriteLine();
        }
    }
}
```

Output

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
Jagged Array:
Row 0: 1 2 3
Row 1: 4 5
Row 2: 6 7 8 9
```

13.	**Implement c# program to list all substrings in a given string.**
Code	

```
using System;
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Develop by Vidhey H. Detroja");
        Console.WriteLine("Enroll no.- 2303421007");
        Console.WriteLine("Enter a string:");
        string inputString = Console.ReadLine();
        Console.WriteLine("Substrings:");
        ListSubstrings(inputString);
    }
    static void ListSubstrings(string str)
    {
        int length = str.Length;
        for (int i = 0; i < length; i++)
        {
            for (int j = i + 1; j <= length; j++)
            {
                string substring = str.Substring(i, j - i);
                Console.WriteLine(substring);
            }
        }
    }
}
```

Output	

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
Enter a string:
Patel
Substrings:
P
Pa
Pat
Pate
Patel
a
at
ate
atel
t
te
tel
e
el
l
```

14.	**Implement c# program to demonstrate the use of following String Methods.** **ToUpper(), ToLOwer(), Concat(), IndexOf(), Substring()**

Code	using System;
	class Program
	{
	static void Main(string[] args)
	{
	string str1 = "Hello";
	string str2 = "World";
	Console.WriteLine("Develop by Vidhey H. Detroja");
	Console.WriteLine("Enroll no.- 2303421007");
	Console.WriteLine("ToUpper() method:");
	Console.WriteLine("str1 in uppercase: " + str1.ToUpper());
	Console.WriteLine("str2 in uppercase: " + str2.ToUpper());
	Console.WriteLine("\nToLower() method:");
	Console.WriteLine("str1 in lowercase: " + str1.ToLower());
	Console.WriteLine("str2 in lowercase: " + str2.ToLower());
	Console.WriteLine("\nConcat() method:");
	string concatenatedString = string.Concat(str1, " ", str2);
	Console.WriteLine("Concatenated string: " + concatenatedString);
	Console.WriteLine("\nIndexOf() method:");
	int index = concatenatedString.IndexOf("World");
	Console.WriteLine("Index of 'World': " + index);
	Console.WriteLine("\nSubstring() method:");
	string subString = concatenatedString.Substring(index);
	Console.WriteLine("Substring from index " + index + ": " + subString);
	}
	}

Output	

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
ToUpper() method:
str1 in uppercase: HELLO
str2 in uppercase: WORLD

ToLower() method:
str1 in lowercase: hello
str2 in lowercase: world

Concat() method:
Concatenated string: Hello World

IndexOf() method:
Index of 'World': 6

Substring() method:
Substring from index 6: World
```

15.	**Implement c# program to demonstrate the use of Access Modifiers.**
Code	

```
using System;
public class Animal
{
    public string Name;
    protected int Age;
    private string Species;
    public Animal(string name, int age, string species)
    {
        Name = name;
        Age = age;
        Species = species;
    }
    public void DisplayInfo()
    {
        Console.WriteLine($"Name: {Name}, Age: {Age}, Species: {Species}");
    }
}
public class Dog : Animal
{


    public Dog(string name, int age) : base(name, age, "Dog")
    {
    }
    public new void DisplayInfo()
    {
        Console.WriteLine($"Dog's Name: {Name}, Dog's Age: {Age}");
    }
}
class Program
```

```
{
    static void Main(string[] args)
    {
        Console.WriteLine("Develop by Vidhey H. Detroja");
        Console.WriteLine("Enroll no.- 2303421007");
        Animal animal = new Animal("goldy", 5, "Cat");
        Console.WriteLine("Accessing public members of Animal class:");
        Console.WriteLine($"Name: {animal.Name}");
        animal.DisplayInfo();
        Console.WriteLine();
        Dog dog = new Dog("Buddy", 3);
        Console.WriteLine("Accessing public members of Dog class:");
        Console.WriteLine($"Name: {dog.Name}");
        dog.DisplayInfo();
    }
}
```

Output

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
Accessing public members of Animal class:
Name: goldy
Name: goldy, Age: 5, Species: Cat

Accessing public members of Dog class:
Name: Buddy
Dog's Name: Buddy, Dog's Age: 3
```

16.	**Implement c# program to demonstrate the use of Constructor and Destructor.**
Code	

```
using System;
public class MyClass
{
    public MyClass()
    {
        Console.WriteLine("Constructor called.");
    }
    ~MyClass()
    {
        Console.WriteLine("Destructor called.");
    }
}
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Develop by Vidhey H. Detroja");
        Console.WriteLine("Enroll no.- 2303421007");
        MyClass obj = new MyClass();
    }
}
```

Output

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
Constructor called.
```

17.	**Implement c# program to demonstrate the use of Inheritance (Single Inheritance, Multi-Level Inheritance)**

Code	using System;
	public class Animal
	{
	public void Eat()
	{
	Console.WriteLine("Animal is eating.");
	}
	}
	public class Dog : Animal
	{
	public void Bark()
	{
	Console.WriteLine("Dog is barking.");
	}
	}
	public class Labrador : Dog
	{
	public void DisplayColor()
	{
	Console.WriteLine("Labrador is brown in color.");
	}
	}
	class Program
	{
	static void Main(string[] args)
	{
	Console.WriteLine("Develop by Vidhey H. Detroja");
	Console.WriteLine("Enroll no.- 2303421007");

```
        Console.WriteLine("Single Inheritance:");

        Dog dog = new Dog();

        dog.Eat();

        dog.Bark();

        Console.WriteLine();

        Console.WriteLine("Multi-Level Inheritance:");

        Labrador labrador = new Labrador();

        labrador.Eat();

        labrador.Bark();

        labrador.DisplayColor();

    }

}
```

Output

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
Single Inheritance:
Animal is eating.
Dog is barking.

Multi-Level Inheritance:
Animal is eating.
Dog is barking.
Labrador is brown in color.
```

18.	**Implement c# program to demonstrate the use of Polymorphism (Method Overloading and Method Overriding)**
Code	

```
using System;
public class Shape
{
    public void Draw()
    {
        Console.WriteLine("Drawing a shape");
    }
    public virtual void CalculateArea()
    {
        Console.WriteLine("Calculating area of a shape");
    }
}
public class Circle : Shape
{
    public void Draw(int diameter)
    {
        Console.WriteLine($"Drawing a circle with diameter {diameter}");
    }
    public override void CalculateArea()
    {
        Console.WriteLine("Calculating area of a circle");
    }
}
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Develop by Vidhey H. Detroja");
```

```
        Console.WriteLine("Enroll no.- 2303421007");

        Shape shape = new Shape();

        shape.Draw();

        Circle circle = new Circle();

        circle.Draw(10);

        Console.WriteLine();

        Shape circleShape = new Circle();

        circleShape.CalculateArea();

    }

}
```

Output

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
Drawing a shape
Drawing a circle with diameter 10

Calculating area of a circle
```

19.	**Implement c# program to demonstrate the use of the following methods of ArrayList Class.** **Add(), Insert(), Remove(), RemoveAt(), Sort(), Reverse(), Contains(), IndexOf()**

Code	using System;
	using System.Collections;
	class Program
	{
	static void Main(string[] args)
	{
	Console.WriteLine("Develop by Vidhey H. Detroja");
	Console.WriteLine("Enroll no.- 2303421007");
	ArrayList arrayList = new ArrayList();
	arrayList.Add("Apple");
	arrayList.Add("Banana");
	arrayList.Add("Orange");
	arrayList.Insert(1, "Grapes");
	Console.WriteLine("Elements in the ArrayList:");
	foreach (var item in arrayList)
	{
	Console.WriteLine(item);
	}
	Console.WriteLine();
	arrayList.Remove("Banana");
	arrayList.RemoveAt(2);
	Console.WriteLine("Elements in the ArrayList after removal:");
	foreach (var item in arrayList){
	Console.WriteLine(item);}
	Console.WriteLine();
	arrayList.Sort();
	Console.WriteLine("Elements in the ArrayList after sorting:");
	foreach (var item in arrayList) {

```
            Console.WriteLine(item);}

        Console.WriteLine();

        arrayList.Reverse();

        Console.WriteLine("Elements in the ArrayList after reversing:");

        foreach (var item in arrayList){

            Console.WriteLine(item);}

        Console.WriteLine();

        Console.WriteLine("Does the ArrayList contain 'Grapes'? " + arrayList.Contains("Grapes"));

        Console.WriteLine("Does the ArrayList contain 'Mango'? " + arrayList.Contains("Mango"));

        Console.WriteLine();

        Console.WriteLine("Index of 'Apple' in the ArrayList: " + arrayList.IndexOf("Apple"));

        Console.WriteLine("Index of 'Orange' in the ArrayList: " + arrayList.IndexOf("Orange"));

    }

}
```

Output

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
Elements in the ArrayList:
Apple
Grapes
Banana
Orange

Elements in the ArrayList after removal:
Apple
Grapes

Elements in the ArrayList after sorting:
Apple
Grapes

Elements in the ArrayList after reversing:
Grapes
Apple

Does the ArrayList contain 'Grapes'? True
Does the ArrayList contain 'Mango'? False

Index of 'Apple' in the ArrayList: 1
Index of 'Orange' in the ArrayList: -1
```

20.	**Implement c# program to demonstrate the use of the following methods of Hashtable Class.** **Add(), Remove(), ContainsKey(), ContainsValue()**

Code	using System;
	using System.Collections;
	class Program
	{
	static void Main(string[] args)
	{
	Console.WriteLine("Develop by Vidhey H. Detroja");
	Console.WriteLine("Enroll no.- 2303421007");
	Hashtable hashtable = new Hashtable();
	hashtable.Add("John", 25);
	hashtable.Add("Alice", 30);
	hashtable.Add("Bob", 28);
	Console.WriteLine("Key-Value pairs in the Hashtable:");
	foreach (DictionaryEntry entry in hashtable)
	{
	Console.WriteLine($"Key: {entry.Key}, Value: {entry.Value}");
	}
	Console.WriteLine();
	hashtable.Remove("Alice");
	Console.WriteLine("Key-Value pairs in the Hashtable after removal:");
	foreach (DictionaryEntry entry in hashtable)
	{
	Console.WriteLine($"Key: {entry.Key}, Value: {entry.Value}");
	}
	Console.WriteLine();
	Console.WriteLine("Does the Hashtable contain key 'Alice'? " + hashtable.ContainsKey("Alice"));
	Console.WriteLine("Does the Hashtable contain key 'Bob'? " +

	hashtable.ContainsKey("Bob"));
	
	Console.WriteLine();
	
	Console.WriteLine("Does the Hashtable contain value 30? " + hashtable.ContainsValue(30));
	
	Console.WriteLine("Does the Hashtable contain value 28? " + hashtable.ContainsValue(28));
	
	}
	
	}

Output	Develop by Vidhey H. Detroja Enroll no.- 2303421007 Key-Value pairs in the Hashtable: Key: John, Value: 25 Key: Alice, Value: 30 Key: Bob, Value: 28 Key-Value pairs in the Hashtable after removal: Key: John, Value: 25 Key: Bob, Value: 28 Does the Hashtable contain key 'Alice'? False Does the Hashtable contain key 'Bob'? True Does the Hashtable contain value 30? False Does the Hashtable contain value 28? True

21.	**Implement c# program to demonstrate the use of Exception Handling. (try, catch, finally)**
Code	

```
using System;
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Develop by Vidhey H. Detroja");
        Console.WriteLine("Enroll no.- 2303421007");
        try
        {
            int[] numbers = { 1, 2, 3, 4, 5 };
            Console.WriteLine("Accessing element at index 10: " + numbers[10]);
        }
        catch (IndexOutOfRangeException ex)
        {
            Console.WriteLine("Exception caught: " + ex.Message);
        }
        finally
        {
            Console.WriteLine("Inside finally block (Example 1)");
        }
        Console.WriteLine();
        try
        {
            int x = 10;
            int y = 0;
            int result = x / y; // Division by zero will throw an exception
            Console.WriteLine("Result of division: " + result);
        }
        catch (DivideByZeroException ex)
```

```
        {
            Console.WriteLine("Exception caught: " + ex.Message);
        }
        finally
        {
            Console.WriteLine("Inside finally block (Example 2)");
        }
        Console.WriteLine();
        try
        {
            string str = null;
            Console.WriteLine("Length of string: " + str.Length);
        }
        finally
        {
            Console.WriteLine("Inside finally block (Example 3)");
        }
    }
}
```

Output

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
Exception caught: Index was outside the bounds of the array.
Inside finally block (Example 1)

Exception caught: Attempted to divide by zero.
Inside finally block (Example 2)
```

22.	**Implement c# program to demonstrate the use of LINQ Query Syntax and LINQ Method Syntax.**
Code	

```
using System;

using System.Collections.Generic;

using System.Linq;

class Program{

    static void Main(string[] args){

        Console.WriteLine("Develop by Vidhey H. Detroja");

        Console.WriteLine("Enroll no.- 2303421007");

        List<int> numbers = new List<int> { 1, 2, 3, 4, 5 };

        var querySyntaxResult = from num in numbers

                                where num % 2 == 0

                                select num;

        Console.WriteLine("Result using LINQ Query Syntax:");

        foreach (var num in querySyntaxResult){

            Console.WriteLine(num);

        }

        Console.WriteLine();

        var methodSyntaxResult = numbers.Where(num => num % 2 == 0);

        Console.WriteLine("Result using LINQ Method Syntax:");

        foreach (var num in methodSyntaxResult){

            Console.WriteLine(num);

        }

    }

}
```

Output	`Develop by Vidhey H. Detroja`
	`Enroll no.- 2303421007`
	`Result using LINQ Query Syntax:`
	`2`
	`4`
	
	`Result using LINQ Method Syntax:`
	`2`
	`4`

23.	**Implement c# program to demonstrate the use of LINQ Query Operators (select, where, OrderBy, Count, Max, Min, Sum, All, Any, Contains, etc.)**
Code	

```
using System;
using System.Collections.Generic;
using System.Linq;
class Program
{
   static void Main(string[] args)
   {
      Console.WriteLine("Develop by Vidhey H. Detroja");
      Console.WriteLine("Enroll no.- 2303421007");
      List<int> numbers = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
      var evenNumbers = from num in numbers
                        where num % 2 == 0
                        select num;
      Console.WriteLine("Even numbers:");
      foreach (var num in evenNumbers)
      {
         Console.WriteLine(num);
      }
      Console.WriteLine();
      var orderedNumbers = from num in numbers
                           orderby num descending
                           select num;
      Console.WriteLine("Numbers in descending order:");
      foreach (var num in orderedNumbers)
      {
         Console.WriteLine(num);
      }
      Console.WriteLine();
```

```
        int count = numbers.Count();
        Console.WriteLine("Count of numbers: " + count);
        Console.WriteLine();
        int maxNumber = numbers.Max();
        Console.WriteLine("Maximum number: " + maxNumber);
        Console.WriteLine();
        int minNumber = numbers.Min();
        Console.WriteLine("Minimum number: " + minNumber);
        Console.WriteLine();
        int sum = numbers.Sum();
        Console.WriteLine("Sum of numbers: " + sum);
        Console.WriteLine();
        bool allEven = numbers.All(num => num % 2 == 0);
        Console.WriteLine("Are all numbers even? " + allEven);
        Console.WriteLine();
        bool anyOdd = numbers.Any(num => num % 2 != 0);
        Console.WriteLine("Are there any odd numbers? " + anyOdd);
        Console.WriteLine();
        bool containsSeven = numbers.Contains(7);
        Console.WriteLine("Does the list contain 7? " + containsSeven);
    }
}
```

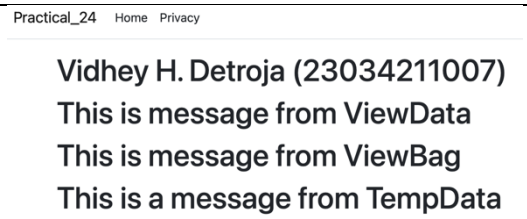
Output

```
Develop by Vidhey H. Detroja
Enroll no.- 2303421007
Even numbers:
2
4
6
8
10

Numbers in descending order:
10
9
8
7
6
5
4
3
2
1

Count of numbers: 10

Maximum number: 10

Minimum number: 1

Sum of numbers: 55

Are all numbers even? False

Are there any odd numbers? True

Does the list contain 7? True
```

24.	**Develop ASP.NET Core MVC web application to demonstrate the use of ViewData, Viewbag and TempData.**

Home Controller.cs	public IActionResult Index()
	{
	ViewData["De"] = "Vidhey H. Detroja (23034211007)";
	ViewData["Vi"] = "This is message from ViewData";
	ViewBag.Message = "This is message from ViewBag";
	TempData["Message"] = "This is a message from TempData";
	return View();
	}

Index.cshtml	@{
	ViewData["Title"] = "Home Page";
	}
	<h1>@ViewData["De"]</h1>
	<h1>@ViewData["Vi"]</h1>
	<h1>@ViewBag.Message</h1>
	<h1>@TempData["Message"]</h1>

Output	

25.	**Develop ASP.NET Core MVC web application to Create a Layout View in ASP.NET Core MVC.**

Index.cshtml	@{
	ViewData["Title"] = "Home Page";
	}
	<h1 class="display-4">Mobile Store</h1>

Layout.cshtml	<!DOCTYPE html>
	<html lang="en">
	<head>
	<meta charset="UTF-8">
	<meta http-equiv="X-UA-Compatible" content="IE=edge">
	<meta name="viewport" content="width=device-width, initial-scale=1.0">
	<title>Mobile Phone Landing Page</title>
	<style>
	/* CSS styles */
	body {
	font-family: Arial, sans-serif;
	margin: 0;
	}
	header {
	background-color: #333;
	color: #fff;
	display: flex;
	justify-content: space-between;
	align-items: center;
	padding: 10px;
	}
	.logo {
	font-size: 1.5em;

```
}
nav ul {
    list-style: none;
    margin: 0;
    padding: 0;
    display: flex;
}
nav ul li {
    margin: 0 10px;
}
nav ul li a {
    color: #fff;
    text-decoration: none;
}
#menu-button {
    display: none;
    font-size: 1.5em;
    background-color: #333;
    color: #fff;
    border: none;
}
.hero {
    padding: 20px;
    text-align: center;
    background-color: #f4f4f4;
}
.hero img {
    width: 80%;
```

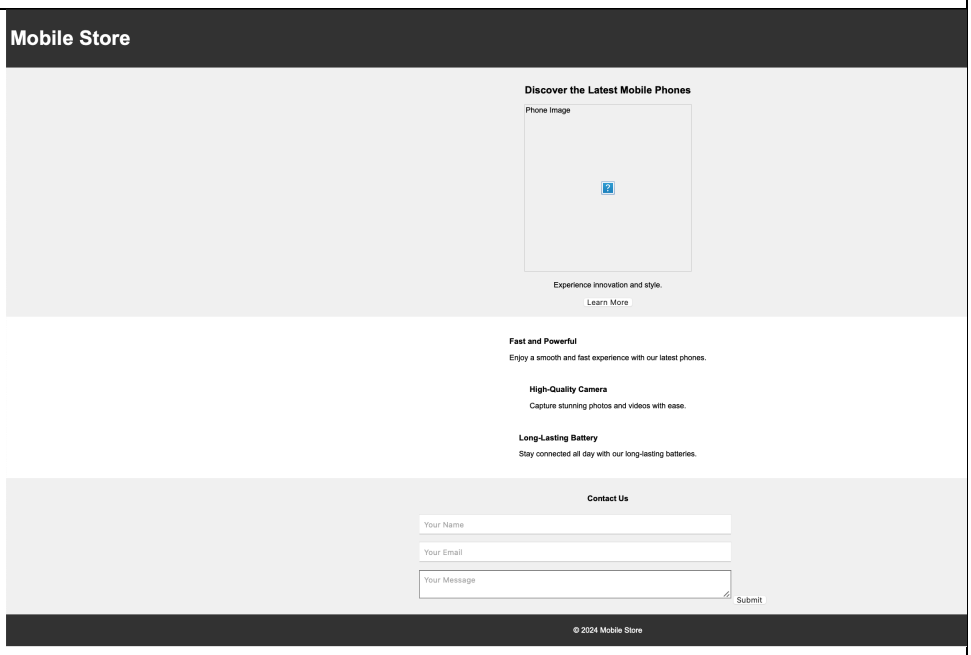
```
    max-width: 400px;
}
.features {
    padding: 20px;
    display: flex;
    flex-direction: column;
    align-items: center;
    background-color: #fff;
}
.features div {
    margin: 10px 0;
}

footer {
    background-color: #333;
    color: #fff;
    text-align: center;
    padding: 10px;
}
.contact-form {
    display: flex;
    flex-direction: column;
    align-items: center;
    padding: 20px;
    background-color: #f4f4f4;
}
.contact-form input,
.contact-form textarea {
```

```
            width: 80%;
            margin: 10px 0;
            padding: 10px;
        }
    </style>
</head>
<body>
    <header>
        <div class="logo">
            @RenderBody()
        </div>
        <nav id="nav-bar">
            <ul>
                <li><a href="#">Home</a></li>
                <li><a href="#">Phones</a></li>
                <li><a href="#">Contact</a></li>
            </ul>
        </nav>
        <button id="menu-button" onclick="toggleMenu()">☰</button>
    </header>
    <main>
        <section class="hero">
            <h2>Discover the Latest Mobile Phones</h2>
            <img src="M/Users/apple/Desktop/V/Classroom/MCA Sem-
2/Practical/WTD/Practical 25/Practical 25/Views/Shared/peakpx.jpg"
alt="Phone Image">
            <p>Experience innovation and style.</p>
            <button onclick="showDetails()">Learn More</button>
        </section>
```

```
        <section class="features">
            <div>
                <h3>Fast and Powerful</h3>
                <p>Enjoy a smooth and fast experience with our latest phones.</p>
            </div>
            <div>
                <h3>High-Quality Camera</h3>
                <p>Capture stunning photos and videos with ease.</p>
            </div>
            <div>
                <h3>Long-Lasting Battery</h3>
                <p>Stay connected all day with our long-lasting batteries.</p>
            </div>
        </section>
        <section class="contact-form">
            <h3>Contact Us</h3>
            <form>
                <input type="text" name="name" placeholder="Your Name"
required>
                <input type="email" name="email" placeholder="Your Email"
required>
                <textarea name="message" placeholder="Your Message"
required></textarea>
                <button type="submit">Submit</button>
            </form>
        </section>
    </main>
    <footer>
        <p>&copy; 2024 Mobile Store</p>
```

```
    </footer>
    <script>function toggleMenu() {
        const navBar = document.getElementById('nav-bar');
        navBar.classList.toggle('show');
    }
    function showDetails() {
        alert('Discover more about our phones on our website!');
    }
    document.querySelector('.contact-form form').addEventListener('submit',
function (event) {
        event.preventDefault();
        alert('Form submitted successfully!');
    });</script>
</body>
</html>
```

Output



26.	**Develop ASP.NET Core MVC web application to demonstrate the use of Razor Syntax (If …else, Switch, For Loop, For each Loop, While Loop)**
Index.cshtml	

```
@{
   Layout = null;
 }
      <center><h1>IF...ELSE</h1></center>
      <center><h1>--------------------</h1></center>
      @{
         int num1 = 10, num2 = 3;
         String num3;
         if (num1 % num2 == 0)
         {
            num3 = "The number is Even!";
         }
         else
         {
            num3 = "The number is Odd!";
         }
      }
      <center><h1>Th<b>@num1</b> divide by <b>@num2</b>
<b>@num3</b></h1></center>


      <center><h1>FOR LOOP</h1></center>
      <center><h1>--------------------</h1></center>
         @for (int i = 0; i <= 2; i++)
         {
            <center><li><h1>@i</h1></li></center>
         }
      <center><h1>WHILE LOOP</h1></center>
```

```
<center><h1>--------------------</h1></center>
<ul>
   @{
      var n = 0;
      while (n < 2)
      {
         n += 1;
         <center><p><h1>Line @n</h1></p></center>
      }
   }
</ul>
<center><h1>FOREACH</h1></center>
<center><h1>--------------------</h1></center>
@{
   string[] members = { "Apple", "Banana", "Grapes" };
   foreach (var person in members)
   {
      <center><p><h1>@person</h1></p></center>
   }
}
<center><h1>SWITCH CASE</h1></center>
<center><h1>--------------------</h1></center>
@switch (DateTime.Now.DayOfWeek)
{
   case DayOfWeek.Monday:
      <center><h1>Uh-oh...</h1></center>
      break;
   case DayOfWeek.Friday:
```

```
            <center><h1>Weekend coming up!</h1></center>
            break;
        case DayOfWeek.Saturday:
            <center><h1>Saturday AA Gaya!</h1></center>
            break;
        case DayOfWeek.Sunday:
            <center><h1>Finally weekend</h1></center>
            break;
        default:
            <center><h1><span>Nothing special about this day...</span></h1></center>
            break;
    }
```

Output

Vidhey H. Detroja

IF...ELSE

Th10 divide by 3 The number is Odd!
FOR LOOP

. 0
. 1
. 2
WHILE LOOP

Line 1
Line 2
FOREACH

Apple
Banana
Grapes
SWITCH CASE

Nothing special about this day...

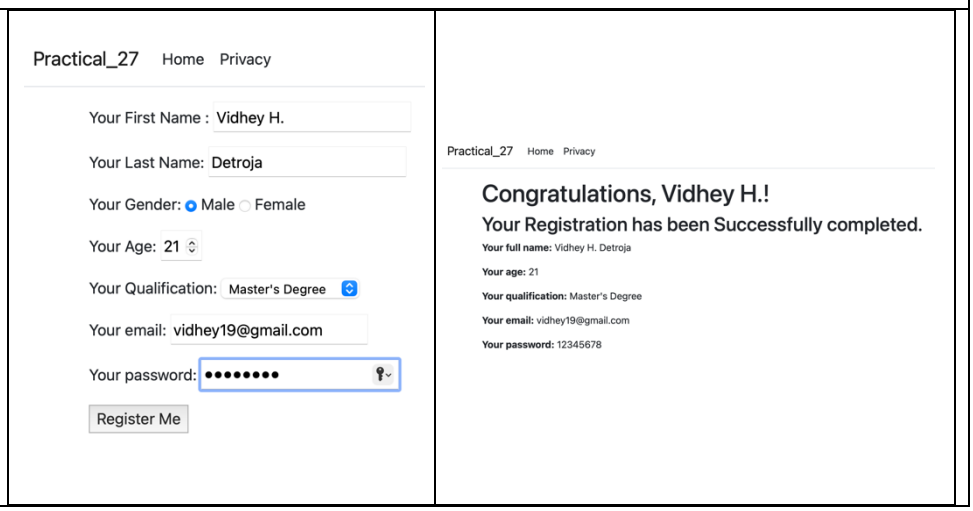
27.	**Create ASP.NET Core MVC registration page using Razor Syntax and Html Helper methods to register new student.** **First name (TextBox)** **Last name or initial (TextBox)** **Gender (Radio Button)** **Age (Take numeric input only)** **Existing Qualification(Dropdown List)** **Email address (Regular Expression)** **Choose a password (min. 8 characters)**

Vidhey.cs	using System;
	
	namespace Practical_27.Models
	
	{
	
	public class Vidhey
	
	{
	
	public string FirstName { get; set; }
	
	public string LastName { get; set; }
	
	public string Gender { get; set; }
	
	public int Age { get; set; }
	
	public string Qualification { get; set; }
	
	public string Email { get; set; }
	
	public string Password { get; set; }
	
	}
	
	}

Home Controller.cs	[HttpGet]
	
	public IActionResult Index()
	
	{
	
	return View();
	
	}
	
	[HttpPost]

	public IActionResult Index(Vidhey registration)
	{
	return View("MyView", registration);
	}

Index.cshtml	@model Practical_27.Models.Vidhey; @using (Html.BeginForm("Index", "Home", FormMethod.Post, new { onsubmit = "return validateForm();" })) { Your First Name : @Html.TextBoxFor(x => x.FirstName, new { id = "FirstName" }) Your Last Name: @Html.TextBoxFor(x => x.LastName, new { id = "LastName" }) Your Gender: @Html.RadioButtonFor(x => x.Gender, "Male", new { id = "GenderMale" }) Male @Html.RadioButtonFor(x => x.Gender, "Female", new { id = "GenderFemale" }) Female Your Age: @Html.TextBoxFor(x => x.Age, new { id = "Age", type = "number", min = "11", max = "24" }) Your Qualification: @Html.DropDownListFor(x => x.Qualification, new SelectList(new List<string> { "High School", "College", "Bachelor's Degree", "Master's Degree", "PhD" }), "Select", new { id = "Qualification" })

	   Your email: @Html.TextBoxFor(x => x.Email, new { id = "Email", type = "email" }) Your password: @Html.PasswordFor(x => x.Password, new { id = "Password", minlength = "3", maxlength = "8" }) <input type="submit" value="Register Me" /> }
MyView.cshtml	@model Practical_27.Models.Vidhey; <h1>Congratulations, @Model.FirstName!</h1> <h2>Your Registration has been Successfully completed.</h2> > Your full name: @Model.FirstName @Model.LastName Your age: @Model.Age Your qualification: @Model.Qualification Your email: @Model.Email Your password: @Model.Password
Output	

28.	**Develop ASP.NET core MVC web application to perform CRUD operation by using entity framework with validation on Book model.** **Book (BookId, BookTitle, AuthorName, Publication and Price)**

Book.cs	using System.ComponentModel.DataAnnotations;
	
	namespace Program28.Models
	
	{
	
	public class Book
	
	{
	
	[Key]
	
	public int BookId { get; set; }
	
	[Required(ErrorMessage = "Title is required")]
	
	public string BookTitle { get; set; }
	
	[Required(ErrorMessage = "Author name is required")]
	
	public string AuthorName { get; set; }
	
	[Required(ErrorMessage = "Publication is required")]
	
	public string Publication { get; set; }
	
	[Required(ErrorMessage = "Price is required")]
	
	[Range(0, double.MaxValue, ErrorMessage = "Price must be a positive number")]
	
	public decimal Price { get; set; }
	
	}
	
	}

BookControl.cs	using System; using System.Collections.Generic; using System.Linq; using System.Threading.Tasks; using Microsoft.AspNetCore.Mvc; using Microsoft.AspNetCore.Mvc.Rendering; using Microsoft.EntityFrameworkCore;

```
using Program28.Data;
using Program28.Models;
namespace Program28.Controllers
{
    public class BooksController : Controller
    {
        private readonly Program28Context _context;

        public BooksController(Program28Context context)
        {
            _context = context;
        }
        public async Task<IActionResult> Index()
        {
            return View(await _context.Book.ToListAsync());
        }
        public async Task<IActionResult> Details(int? id)
        {
            if (id == null)
            {
                return NotFound();
            }
            var book = await _context.Book
                .FirstOrDefaultAsync(m => m.BookId == id);
            if (book == null)
            {
                return NotFound();
            }
```

```
            return View(book);
        }
        public IActionResult Create()
        {
            return View();
        }
http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<IActionResult>
Create([Bind("BookId,BookTitle,AuthorName,Publication,Price")] Book
book)
        {
            if (ModelState.IsValid)
            {
                _context.Add(book);
                await _context.SaveChangesAsync();
                return RedirectToAction(nameof(Index));
            }
            return View(book);
        }
        public async Task<IActionResult> Edit(int? id)
        {
            if (id == null)
            {
                return NotFound();
            }
            var book = await _context.Book.FindAsync(id);
            if (book == null)
```

```
            {
                return NotFound();
            }
            return View(book);
        }
http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<IActionResult> Edit(int id,
[Bind("BookId,BookTitle,AuthorName,Publication,Price")] Book book)
        {
            if (id != book.BookId)
            {
                return NotFound();
            }
            if (ModelState.IsValid)
            {
                try
                {
                    _context.Update(book);
                    await _context.SaveChangesAsync();
                }
                catch (DbUpdateConcurrencyException)
                {
                    if (!BookExists(book.BookId))
                    {
                        return NotFound();
                    }
```

```
                else
                {
                    throw;
                }
            }
            return RedirectToAction(nameof(Index));
        }
        return View(book);
    }
    public async Task<IActionResult> Delete(int? id)
    {
        if (id == null)
        {
            return NotFound();
        }
        var book = await _context.Book
            .FirstOrDefaultAsync(m => m.BookId == id);
        if (book == null)
        {
            return NotFound();
        }

        return View(book);
    }
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> DeleteConfirmed(int id)
    {
```

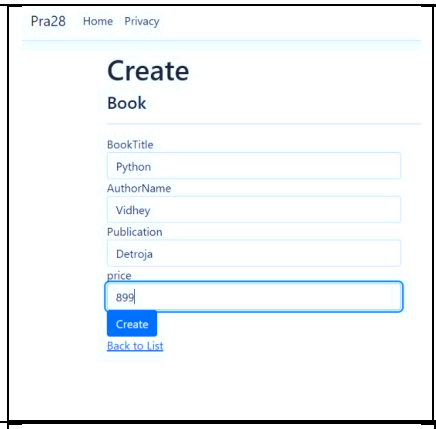
	var book = await _context.Book.FindAsync(id);
	if (book != null)
	{
	_context.Book.Remove(book);
	}
	await _context.SaveChangesAsync();
	return RedirectToAction(nameof(Index));
	}
	private bool BookExists(int id)
	{
	return _context.Book.Any(e => e.BookId == id);
	}
	}
	}

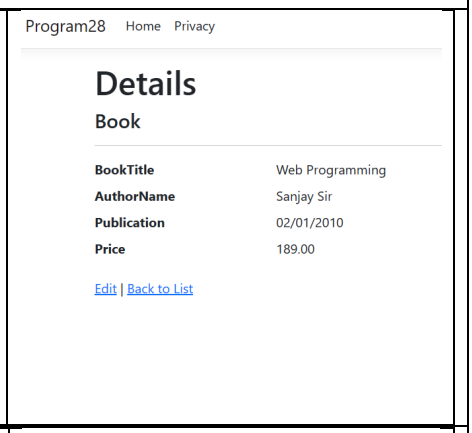
Program28Context.cs	using System; using System.Collections.Generic; using System.Linq; using System.Threading.Tasks; using Microsoft.EntityFrameworkCore; using Program28.Models; namespace Program28.Data { public class Program28Context : DbContext { public Program28Context (DbContextOptions<Program28Context> options) : base(options) {

```
        }

        public DbSet<Program28.Models.Book> Book { get; set; } =
default!;

    }

}
```

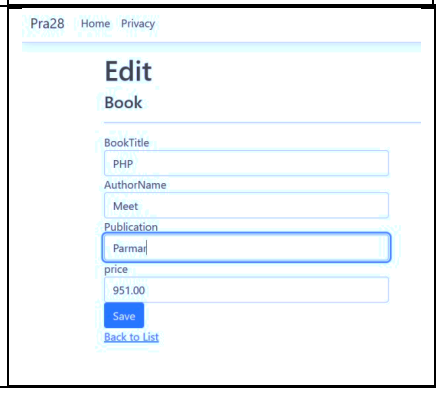
Output

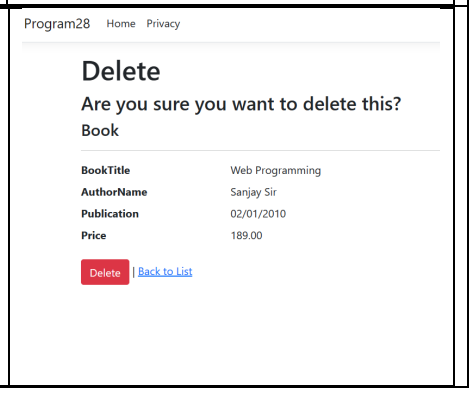
Output





Output





29.	**Develop ASP.NET Core MVC application to perform CRUD operation by using entity framework with validation on Student, Course and Enrollment models.**

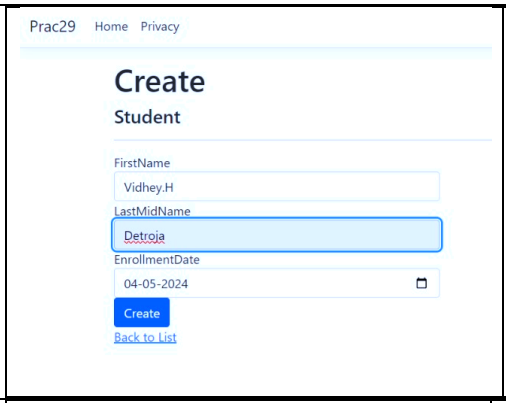
Student.cs	using System;
	using System.Collections.Generic;
	using System.ComponentModel.DataAnnotations;
	namespace CRUD.Models
	{
	public class Student
	{
	[Key]
	public int StudentID { get; set; }
	[Required(ErrorMessage = "First Name is required")]
	[StringLength(50)]
	public string FirstName { get; set; }
	[Required(ErrorMessage = "Last Name is required")]
	[StringLength(50)]
	public string LastMidName { get; set; }
	[DataType(DataType.Date)]
	[DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}",
	ApplyFormatInEditMode = true)]
	public DateTime EnrollmentDate { get; set; }
	public ICollection<Enrollment> Enrollments { get; set; }
	public Student()
	{
	Enrollments = new List<Enrollment>();
	}
	}
	}

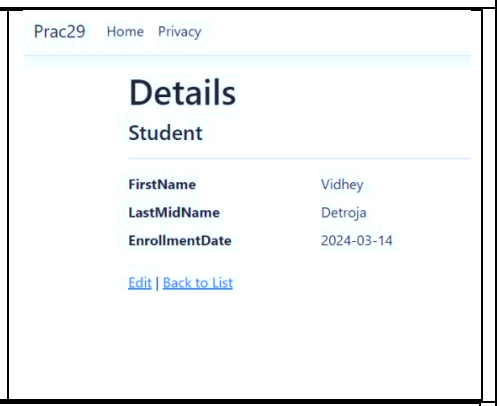
Course.cs	using System.Collections.Generic; using System.ComponentModel.DataAnnotations; namespace CRUD.Models { public class Course { [Key] public int CourseID { get; set; } [Required(ErrorMessage = "Course Title is required")] [StringLength(100, ErrorMessage = "Title cannot exceed 100 characters")] public string CourseTitle { get; set; } [Required(ErrorMessage = "Credits are required")] [Range(1, int.MaxValue, ErrorMessage = "Credits must be a positive number")] public int CourseCredits { get; set; } public ICollection<Enrollment> Enrollments { get; set; } public Course() { Enrollments = new List<Enrollment>(); } }}
Enrollment.c	using System.ComponentModel.DataAnnotations.Schema;

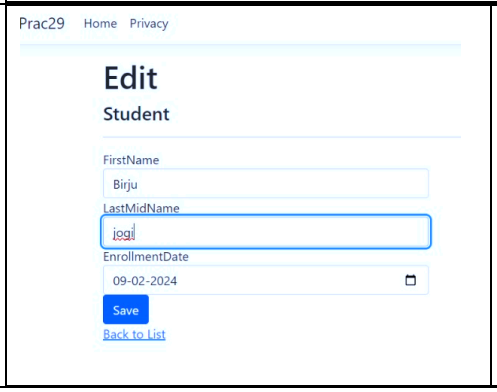
s

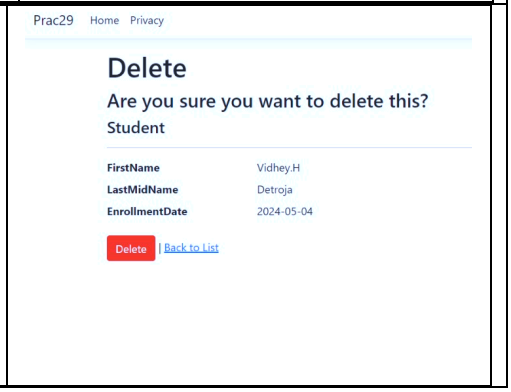
```
using System.ComponentModel.DataAnnotations;
namespace CRUD.Models
{
   public class Enrollment
   {
      [Key]
      public int EnrollmentID { get; set; }
      [Required(ErrorMessage = "CourseID is required")]
      public int CourseID { get; set; }
      [Required(ErrorMessage = "StudentID is required")]
      public int StudentID { get; set; }
      [DataType(DataType.Date)]
      [DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}",
ApplyFormatInEditMode = true)]
      public DateTime EnrollmentDate { get; set; }
      [ForeignKey("CourseID")]
      public Course? Course { get; set; }
      [ForeignKey("StudentID")]
      public Student? Student { get; set; }
   }}
```

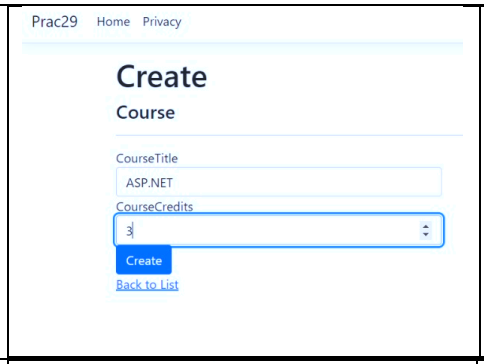
Output

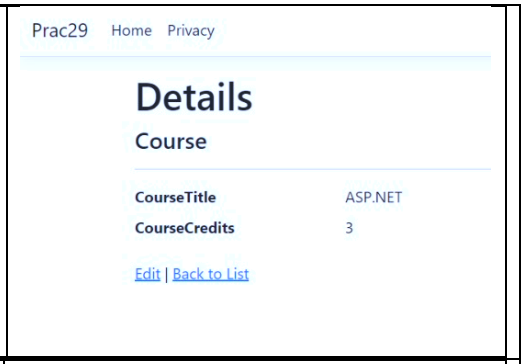


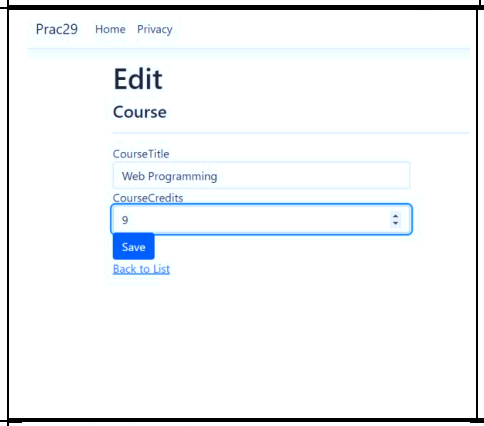


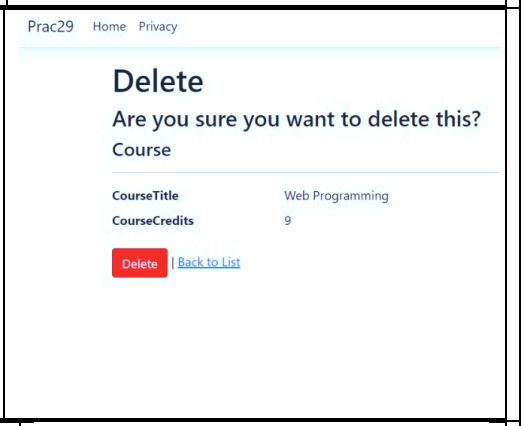


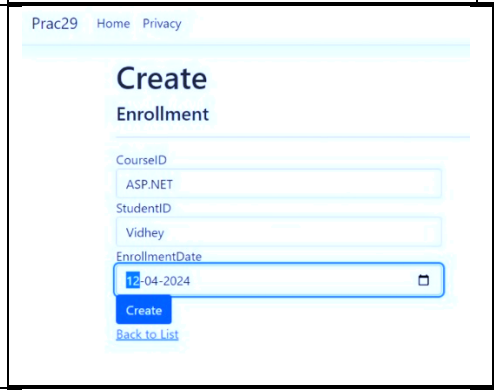


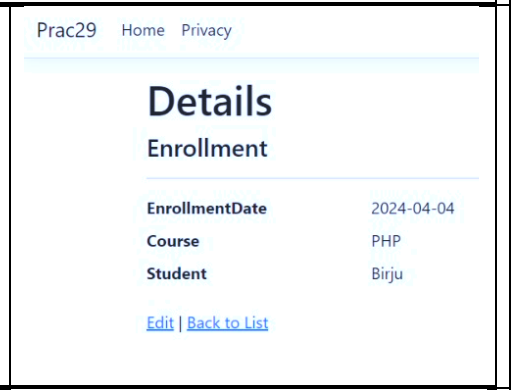

Prac29 Home Privacy
Create
Course
CourseTitle
ASP.NET
CourseCredits
3
Create
Back to List

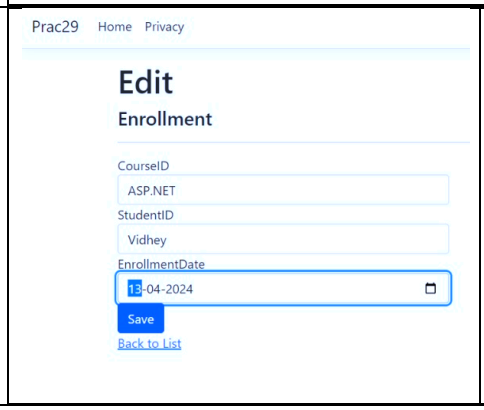

Prac29 Home Privacy
Details
Course
CourseTitle ASP.NET
CourseCredits 3
Edit | Back to List

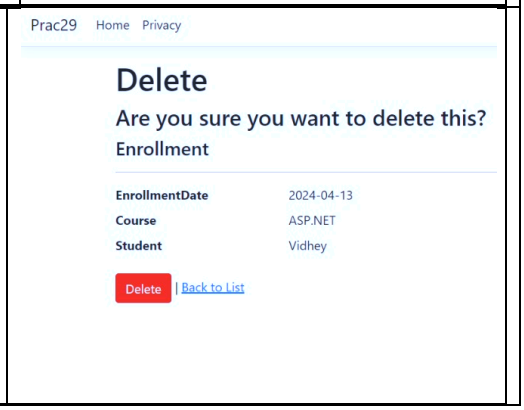

Prac29 Home Privacy
Edit
Course
CourseTitle
Web Programming
CourseCredits
9
Save
Back to List


Prac29 Home Privacy
Delete
Are you sure you want to delete this?
Course
CourseTitle Web Programming
CourseCredits 9
Delete | Back to List


Prac29 Home Privacy
Create
Enrollment
CourseID
ASP.NET
StudentID
Vidhey
EnrollmentDate
12-04-2024
Create
Back to List


Prac29 Home Privacy
Details
Enrollment
EnrollmentDate 2024-04-04
Course PHP
Student Birju
Edit | Back to List


Prac29 Home Privacy
Edit
Enrollment
CourseID
ASP.NET
StudentID
Vidhey
EnrollmentDate
13-04-2024
Save
Back to List


Prac29 Home Privacy
Delete
Are you sure you want to delete this?
Enrollment
EnrollmentDate 2024-04-13
Course ASP.NET
Student Vidhey
Delete | Back to List

<table>
<tr><td></td><td>
Prac29 Home Privacy

Index

Create New

CourseTitle	CourseCredits	
Web Programming	4	Edit | Details | Delete
Software Testing	5	Edit | Details | Delete
ASP.NET	3	Edit | Details | Delete
PHP	6	Edit | Details | Delete
</td></tr>
<tr><td></td><td>
Prac29 Home Privacy

Index

Create New

FirstName	LastMidName	EnrollmentDate	
Vidhey	Detroja	2024-03-14	Edit | Details | Delete
Mihir	Moghariya	2024-02-01	Edit | Details | Delete
Vidhey.H	Detroja	2024-05-04	Edit | Details | Delete
Birju	bitu	2024-02-09	Edit | Details | Delete
</td></tr>
</table>