DEPARTMENT *Computer Science & Information Technology*	NEPATHYA COLLEGE, JANAKINAGAR, BUTWAL	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : Largest of three numbers.	
PAPER CODE: *CSE-314N*		***ISSUE DATE : 1ST Jan, 2018***
EXPERIMENT NO. : 01		***SEMESTER : 6TH Semester***
LABORATORY : *NET CENTRIC COMPUTING LAB*		***PAGE NO. : 01-02***

AIM : Write a C# program to find the largest of three numbers

PROGRAM :

```
using System;
public class LargestNumber
{
   public static void Main()
{
   int num1, num2, num3;
   Console.Write("\n\n");
   Console.Write("Find the largest of three numbers:\n");
   Console.Write("-----------------------------------");
   Console.Write("\n\n");

   Console.Write("Input the 1st number :");
   num1 = Convert.ToInt32(Console.ReadLine());
   Console.Write("Input the  2nd number :");
   num2 = Convert.ToInt32(Console.ReadLine());
   Console.Write("Input the 3rd  number :");
   num3 = Convert.ToInt32(Console.ReadLine());

 if (num1 > num2)
   {
      if (num1 > num3)
      {
         Console.Write("The 1st Number is the greatest among three. \n\n");
      }
      else
      {
         Console.Write("The 3rd Number is the greatest among three. \n\n");
```

```
        }
    }
    else if (num2 > num3)
        Console.Write("The 2nd Number is the greatest among three \n\n");
    else
        Console.Write("The 3rd Number is the greatest among three \n\n");
}
}
```

OUTPUT:

Find the largest of three numbers:

Input the 1st number : 20
Input the 2nd number : 25
Input the 3rd number :15

The 2nd Number is the greatest amount three

DEPARTMENT *Computer Science & Information Technology*	NEPATHYA COLLEGE, JANAKINAGAR, BUTWAL	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : Selection Sort	
PAPER CODE: *CSE-314N*		***ISSUE DATE : 1ST Jan, 2018***
EXPERIMENT NO. : *02*		***SEMESTER : 6TH Semester***
LABORATORY : *NET CENTRIC COMPUTING LAB*		***PAGE NO. : 03-07***

AIM: Write a C# Sharp program to sort a list of elements using the Selection sort algorithm

PROGRAM :

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

namespace Selection_Sort
{
   class Program
   {
      static void Main(string[] args)
      {
         Selection_Sort selection = new Selection_Sort(10);
         selection.Sort();
      }
   }

   class Selection_Sort
   {
      private int[] data;
      private static Random generator = new Random();
      //Create an array of 10 random numbers
        public Selection_Sort(int size)
      {
         data = new int[size];
```

```
        for (int i = 0; i < size; i++)
        {
            data[i] = generator.Next(20, 90);
        }
    }

    public void Sort()
    {
        Console.Write("\nSorted Array Elements :(Step by Step)\n\n");
                        display_array_elements();
                        int smallest;
        for (int i = 0; i < data.Length - 1; i++)
        {
            smallest = i;

            for (int index = i + 1; index < data.Length; index++)
            {
                if (data[index] < data[smallest])
                {
                    smallest = index;
                }
            }
            Swap(i, smallest);
            display_array_elements();
                    }

    }

    public void Swap(int first, int second)
    {
        int temporary = data[first];
        data[first] = data[second];
        data[second] = temporary;
    }

    public void display_array_elements()
    {
        foreach (var element in data)
        {
            Console.Write(element + " ");
        }
        Console.Write("\n\n");
    }
  }
}
```

OUTPUT:

50 70 50 80 89 25 78 58 83 73

25 70 50 80 89 50 78 58 83 73

25 50 70 80 89 50 78 58 83 73

25 50 50 80 89 70 78 58 83 73

25 50 50 58 89 70 78 80 83 73

25 50 50 58 70 89 78 80 83 73

25 50 50 58 70 73 78 80 83 89

25 50 50 58 70 73 78 80 83 89

DEPARTMENT *Computer Science & Information Technology*	NEPATHYA COLLEGE, JANAKINAGAR , BUTWAL	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : Reverse String	
PAPER CODE: *CSE-314N*		***ISSUE DATE :*** *1ST Jan, 2018*
EXPERIMENT NO. : *03*		***SEMESTER :*** *6TH Semester*
LABORATORY : *NET CENTRIC COMPUTING LAB*		***PAGE NO. :*** *08-12*

AIM: Write a program to print individual character of string in reverse order.

PROGRAM :

```
using System;
public class ReverseString
{
   public static void Main()
      {
         string str;
         int l=0;

    Console.Write("\n\nprint individual characters of string in reverse order :\n");
    Console.Write("-----------------------------------------------------\n");
    Console.Write("Input the string : ");
    str = Console.ReadLine();

    l = str.Length - 1;
          Console.Write("The characters of the string in reverse are : \n");
         while (l >= 0)
         {
            Console.Write("{0} ", str[l]);
            l--;
         }
            Console.Write("\n\n");
      }
}
```

OUTPUT:

Print individual characters of string in reverse order

Input the string : ayhtapen

The character of the string in reverse are :

nepathya

DEPARTMENT *Computer Science & Information Technology*	NEPATHYA COLLEGE, JANAKINAGAR , BUTWAL	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : Frequency of Number.	
PAPER CODE: *CSE-314N*		***ISSUE DATE : 1ST Jan, 2018***
EXPERIMENT NO. : 04		***SEMESTER : 6TH Semester***
LABORATORY : *NET CENTRIC COMPUTING LAB*		***PAGE NO. : 13-14***

AIM : Write a program in C# Sharp to display the number and frequency of number from giving array.

PROGRAM :

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

class LinqExercise4
{
   static void Main(string[] args)
   {
       int[] arr1 = new int[] { 5, 9, 1, 2, 3, 7, 5, 6, 7, 3, 7, 6, 8, 5, 4, 9, 6, 2 };
       Console.Write("\nLINQ : Display the number and frequency of number from given array :
\n");
       Console.Write("------------------------------------------------------------------\n");
       Console.Write("The numbers in the array  are : \n");
       Console.Write(" 5, 9, 1, 2, 3, 7, 5, 6, 7, 3, 7, 6, 8, 5, 4, 9, 6, 2\n");

                var n = from x in arr1
                            group x by x into y
                            select y;
                            Console.WriteLine("\nThe number and the Frequency are : \n");
                        foreach (var arrNo in n)
                            {
                                Console.WriteLine("Number "+arrNo.Key + " appears " +
arrNo.Count()+" times");
                            }
      Console.WriteLine("\n");
   }}
```

OUTPUT:

LINQ : Display the number and frequency of number from given array :

The numbers in the array are :
5, 9, 1, 2, 3, 7, 5, 6, 7, 3, 7, 6, 8, 5, 4, 9, 6, 2
The number and the Frequency are :
Number 5 appears 3 times
Number 9 appears 2 times
Number 1 appears 1 times
Number 2 appears 2 times
Number 3 appears 2 times
Number 7 appears 3 times
Number 6 appears 3 times
Number 8 appears 1 times
Number 4 appears 1 times

DEPARTMENT *Computer Science & Information Technology*	NEPATHYA COLLEGE, JANAKINAGAR , BUTWAL	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : Append String on file	
PAPER CODE: *CSE-314N*		***ISSUE DATE :*** *1ST Jan, 2018*
EXPERIMENT NO. : *05*		***SEMESTER :*** *6TH Semester*
LABORATORY : *NET CENTRIC COMPUTING LAB*		***PAGE NO. :*** *15-17*

AIM : Write a program in C# Sharp to create and write some line of text into a file which does not contain a given string in a line.

PROGRAM :

```
using System;
using System.IO;


class WriteTextFile
{
   static void Main()
   {
                string fileName = @"mytest.txt";
                string[] ArrLines;
                string str;

                int n,i;

            Console.Write("\n\n Create and write some line of text which does not contain a given
string in a line  :\n");
                Console.Write("-----------------------------------------------------------------------------
-------\n");

          if (File.Exists(fileName))
          {
             File.Delete(fileName);
          }
       Console.Write(" Input the string to ignore the line : ");
       str = Console.ReadLine();
      Console.Write(" Input number of lines to write in the file  : ");
      n= Convert.ToInt32(Console.ReadLine());
```

```
    ArrLines=new string[n];

    Console.Write(" Input {0} strings below :\n",n);
                for(i=0;i<n;i++)
                {
                Console.Write(" Input line {0} : ",i+1);
                ArrLines[i] = Console.ReadLine();
                }

     using (System.IO.StreamWriter file =
        new System.IO.StreamWriter(@"mytest.txt"))
     {
        foreach (string line in ArrLines)
        {
           if (!line.Contains(str)) // write the line to the file If it doesn't contain the string in str
           {
              file.WriteLine(line);
           }
        }
     }

        using (StreamReader sr = File.OpenText(fileName))
     {
        string s = "";
        Console.Write("\n The line has ignored which contain the string '{0}'. \n",str);
        Console.Write("\n The content of the file is  :\n",n);
        Console.Write("---------------------------------\n");
        while ((s = sr.ReadLine()) != null)
        {
           Console.WriteLine(" {0} ",s);
        }
        Console.WriteLine();
     }
   }
}
```

OUTPUT:

```
Input the string to ignore the line : easy
 Input number of lines to write in the file  : 2
 Input 2 strings below :
 Input line 1 : nepathya
 Input line 2 : it is easy tutorial

 The line has ignored which contain the string 'easy'.

 The content of the file is  :
---------------------------------
 nepathya
```

DEPARTMENT *Computer Science & Information Technology*	NEPATHYA COLLEGE, JANAKINAGAR , BUTWAL	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : CREATE AND READ WITH ASP .NET CORE	
PAPER CODE: *CSE-314N*		***ISSUE DATE :*** *1ST Jan, 2018*
EXPERIMENT NO. : *06*		***SEMESTER :*** *6TH Semester*
LABORATORY : *NET CENTRIC COMPUTING LAB*		***PAGE NO. :*** *18-19*

AIM : Write a .NET CORE program to create object and save into database and read from database

PROGRAM :

Repository.cs

```
using System.Collections.Generic;

namespace FirstApp.Models
{
   public static class Repository
   {
      private static List<Employee> allEmpoyees = new List<Employee>();
      public static IEnumerable<Employee> AllEmpoyees
      {
         get { return allEmpoyees; }
      }
      public static void Create(Employee employee)
      {
         allEmpoyees.Add(employee);
      }
   }
}
```

Employee.cs

```
using System;

namespace FirstApp.Models
{
   public class Employee
   {
      public string Name { get; set; }
      public int Age { get; set; }
      public decimal Salary { get; set; }
      public string Department { get; set; }
      public Char Sex { get; set; }
   }
}
```

Create.cshtml

```
@model Employee
@{
   ViewData["Title"] = "Create";
}

<h2>Create</h2>
<form method="post">
   <p>
      <label asp-for="Name">Your Name:</label>
      <input asp-for="Name" />
   </p>
   <p>
      <label asp-for="Age">Your Age:</label>
      <input asp-for="Age" />
   </p>
   <p>
      <label asp-for="Salary">Your Salary:</label>
      <input asp-for="Salary" />
   </p>
   <p>
      <label asp-for="Department">Your Department:</label>
      <select asp-for="Department">
         <option value="Development">Development</option>
         <option value="HR">HR</option>
         <option value="Research">Research</option>
      </select>
   </p>
   <p>
```

```
        <label asp-for="Sex">Your Sex:</label>
        <input type="radio" asp-for="Sex" value="M" />Male
        <input type="radio" asp-for="Sex" value="F" />Female
    </p>
    <p><button type="submit">Submit</button></p>
</form>
```

EmployeeController.cs

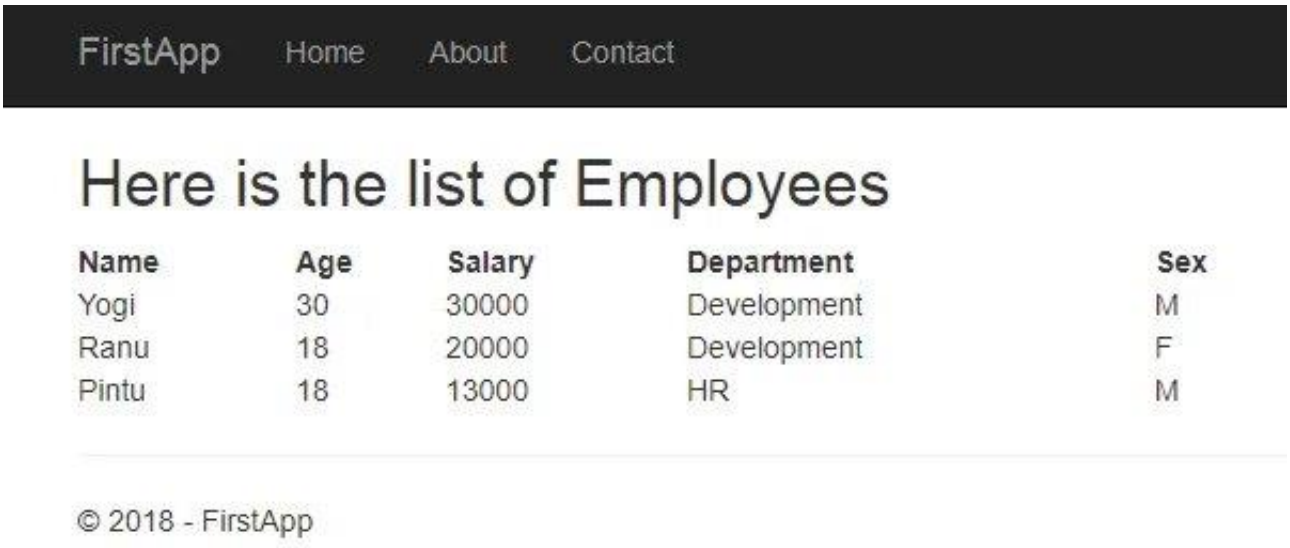
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using FirstApp.Models;
using Microsoft.AspNetCore.Mvc;

namespace FirstApp.Controllers
{
    public class EmployeeController : Controller
    {
        public IActionResult Index()
        {
            return View();
        }

          //Reads all data from database
          public IActionResult Index()
      {
            return View(Repository.AllEmpoyees);
      }


        // HTTP GET VERSION
        public IActionResult Create()
        {
            return View();
        }

        // HTTP POST VERSION
        [HttpPost]
        public IActionResult Create(Employee employee)
        {
            Repository.Create(employee);
            return View("Thanks", employee);
        }
```

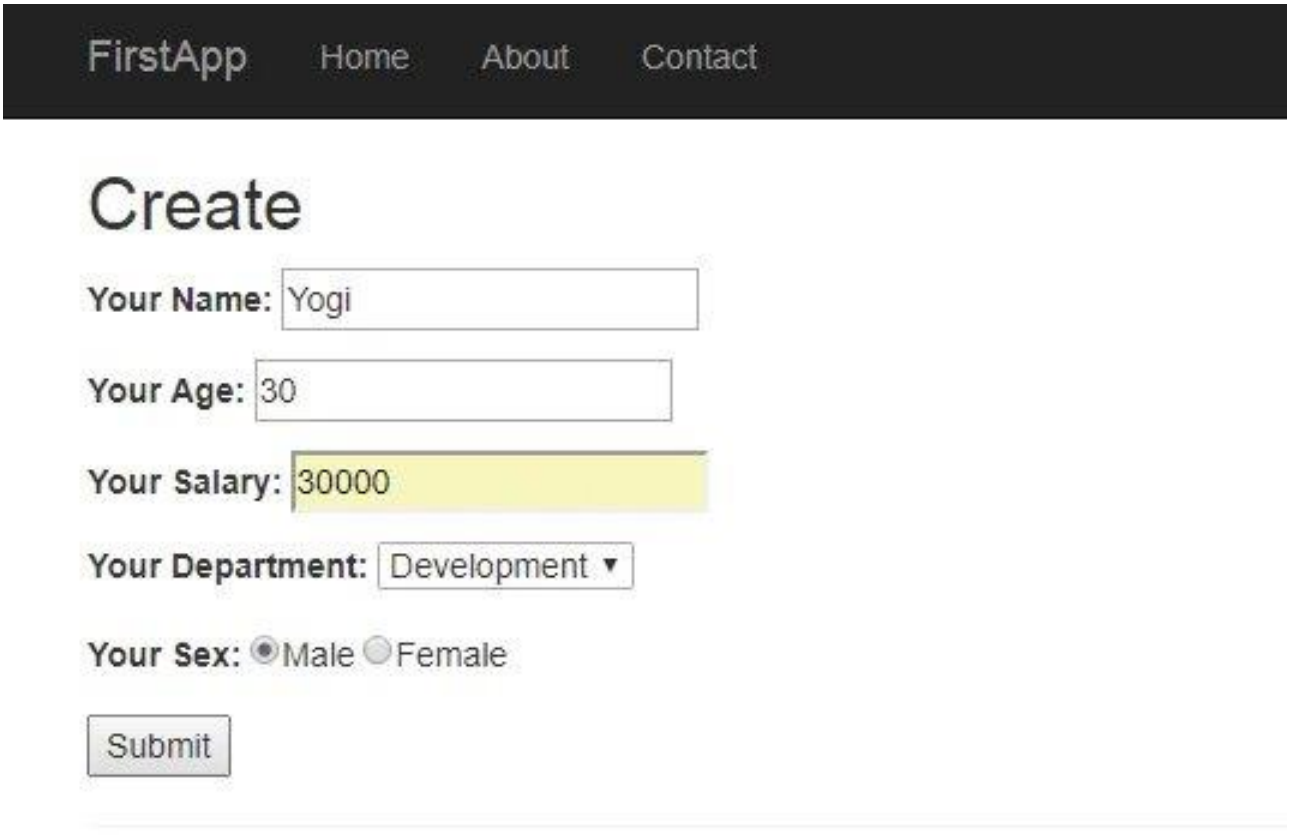
```
    }
}
```

Employee.cshtml

```
@model IEnumerable<Employee>
@{
    ViewData["Title"] = "Index";
}

<h2>Here is the list of Employees</h2>
<table style="width:50%">
    <thead>
        <tr>
            <th>Name</th>
            <th>Age</th>
            <th>Salary</th>
            <th>Department</th>
            <th>Sex</th>
        </tr>
    </thead>
    <tbody>
        @foreach (Employee e in Model)
        {
            <tr>
                <td>@e.Name</td>
                <td>@e.Age</td>
                <td>@e.Salary</td>
                <td>@e.Department</td>
                <td>@e.Sex</td>
            </tr>
        }
    </tbody>
</table>
```

OUTPUT:


FirstApp
Home
About
Contact
Here is the list of Employees
Name Age Salary Department Sex
Yogi 30 30000 Development M
Ranu 18 20000 Development F
Pintu 18 13000 HR M
© 2018 - FirstApp


FirstApp
Home
About
Contact
Create
Your Name: Yogi
Your Age: 30
Your Salary: 30000
Your Department: Development
Your Sex: Male Female
Submit

DEPARTMENT *Computer Science & Information Technology*	NEPATHYA COLLEGE, JANAKINAGAR , BUTWAL	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : Models: Binding and Validations	
PAPER CODE: *CSE-314N*		***ISSUE DATE : 1ST Jan, 2018***
EXPERIMENT NO. : *07*		***SEMESTER : 6TH Semester***
LABORATORY : *NET CENTRIC COMPUTING LAB*		***PAGE NO. : 20-22***

AIM : Write a program to demonstrate the binding and validations.

PROGRAM :

Student.cs

```
using System;
using System.ComponentModel.DataAnnotations;

namespace MvcCoreModelValidation_Demo.Models
{
   public class Student
   {
      [Key]
      public int Id { get; set; }

      [Required(ErrorMessage = "Please enter name")]
      [StringLength(100)]
      public string Name { get; set; }

      [Required(ErrorMessage = "Please choose gender")]
      public string Gender { get; set; }

      [Required(ErrorMessage = "Please enter date of birth")]
      [Display(Name = "Date of Birth")]
      [DataType(DataType.Date)]
      public DateTime DateofBirth { get; set; }
```

```
        [Required(ErrorMessage = "Choose batch time")]
        [Display(Name = "Batch Time")]
        [DataType(DataType.Time)]
        public DateTime BatchTime { get; set; }

        [Required(ErrorMessage = "Please enter phone number")]
        [Display(Name = "Phone Number")]
        [Phone]
        public string PhoneNumber { get; set; }

        [Required(ErrorMessage = "Please enter email address")]
        [Display(Name = "Email Address")]
        [EmailAddress]
        public string Email { get; set; }

        [Required(ErrorMessage = "Please enter website url")]
        [Display(Name = "Website Url")]
        [Url]
        public string WebSite { get; set; }

        [Required(ErrorMessage = "Please enter password")]
        [DataType(DataType.Password)]
        public string Password { get; set; }

        [Required(ErrorMessage = "Please enter confirm password")]
        [Display(Name = "Confirm Password")]
        [Compare("Password", ErrorMessage = "Password and confirm password does not match")]
        public string ConfirmPassword { get; set; }
    }
}
```

Controller.cs

```
using Microsoft.AspNetCore.Mvc;
using MvcCoreModelValidation_Demo.Models;

namespace MvcCoreModelValidation_Demo.Controllers
{
    public class HomeController : Controller
    {
        public IActionResult Index()
        {
            return View();
        }
```

```
        [HttpPost]
        [ValidateAntiForgeryToken]
        public IActionResult Index(Student student)
        {
            if (ModelState.IsValid)
            {

            }
            return View();
        }

    }
}
```

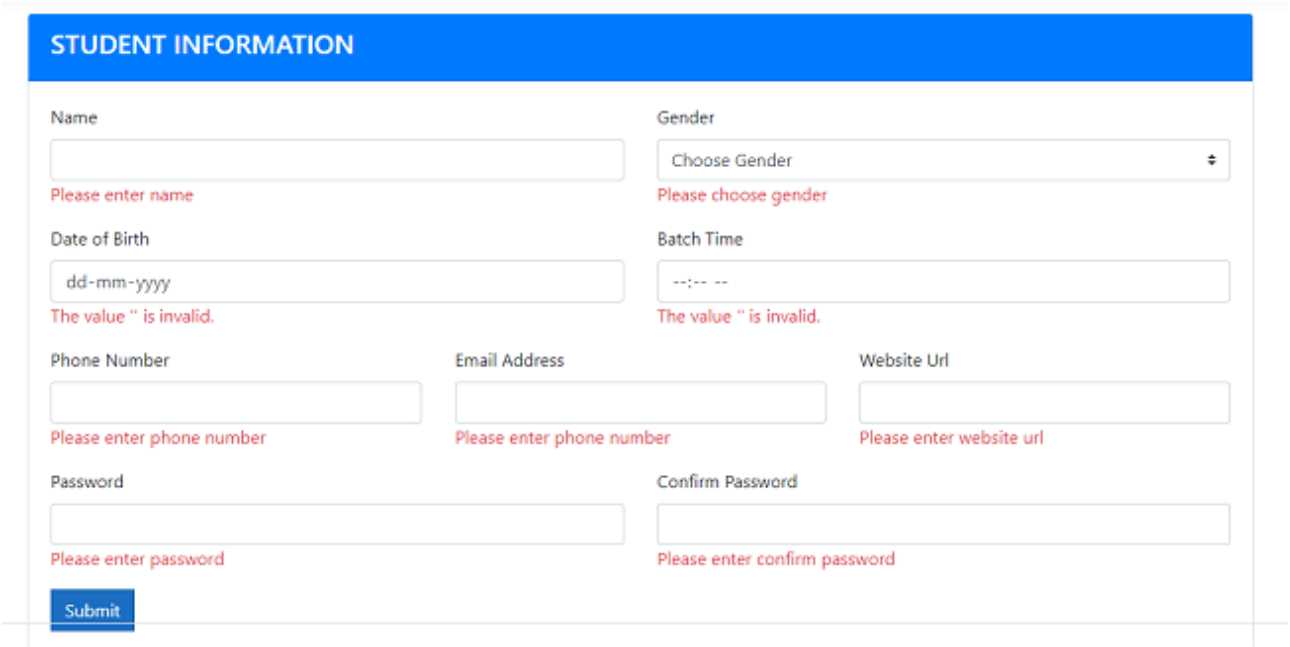
Index.cshtml

```
@model MvcCoreModelValidation_Demo.Models.Student

@{
    ViewData["Title"] = "Home Page";
}

<div class="card">
    <div class="card-header bg-primary text-white text-uppercase">
        <h4>Student Information</h4>
    </div>
    <div class="card-body">
        <form asp-action="Index">
            <div class="row">
                <div class="col-md-6">
                    <div class="form-group">
                        <label asp-for="Name" class="lable-control"></label>
                        <input asp-for="Name" class="form-control" />
                        <span asp-validation-for="Name" class="text-danger"></span>
                    </div>
                </div>
                <div class="col-md-6">
                    <div class="form-group">
                        <label asp-for="Gender" class="lable-control"></label>
                        <select class="custom-select">
                            <option value="">Choose Gender</option>
                            <option value="Male">Male</option>
                            <option value="Female">Female</option>
                        </select>
                        <span asp-validation-for="Gender" class="text-danger"></span>
```

```
        </div>
      </div>
    </div>
    <div class="row">
      <div class="col-md-6">
        <div class="form-group">
          <label asp-for="DateofBirth" class="lable-control"></label>
          <input asp-for="DateofBirth" class="form-control" />
          <span asp-validation-for="DateofBirth" class="text-danger"></span>
        </div>
      </div>
      <div class="col-md-6">
        <div class="form-group">
          <label asp-for="BatchTime" class="lable-control"></label>
          <input asp-for="BatchTime" class="form-control" />
          <span asp-validation-for="BatchTime" class="text-danger"></span>
        </div>
      </div>
    </div>
    <div class="row">
      <div class="col-md-4">
        <div class="form-group">
          <label asp-for="PhoneNumber" class="lable-control"></label>
          <input asp-for="PhoneNumber" class="form-control" />
          <span asp-validation-for="PhoneNumber" class="text-danger"></span>
        </div>
      </div>
      <div class="col-md-4">
        <div class="form-group">
          <label asp-for="Email" class="lable-control"></label>
          <input asp-for="Email" class="form-control" />
          <span asp-validation-for="Email" class="text-danger"></span>
        </div>
      </div>
      <div class="col-md-4">
        <div class="form-group">
          <label asp-for="WebSite" class="lable-control"></label>
          <input asp-for="WebSite" class="form-control" />
          <span asp-validation-for="WebSite" class="text-danger"></span>
        </div>
      </div>
    </div>
    <div class="row">
      <div class="col-md-6">
        <div class="form-group">
          <label asp-for="Password" class="lable-control"></label>
```

```
                <input asp-for="Password" class="form-control" />
                <span asp-validation-for="Password" class="text-danger"></span>
              </div>
            </div>
            <div class="col-md-6">
              <div class="form-group">
                <label asp-for="ConfirmPassword" class="lable-control"></label>
                <input asp-for="ConfirmPassword" class="form-control" />
                <span asp-validation-for="ConfirmPassword" class="text-danger"></span>
              </div>
            </div>
          </div>
          <div class="form-group">
            <button type="submit" class="btn btn-primary rounded-0">Submit</button>
          </div>
        </form>
    </div>
</div>
```

OUTPUT:


STUDENT INFORMATION
Name
Please enter name
Gender
Choose Gender
Please choose gender
Date of Birth
dd-mm-yyyy
The value '' is invalid.
Batch Time
--:-- --
The value '' is invalid.
Phone Number
Please enter phone number
Email Address
Please enter phone number
Website Url
Please enter website url
Password
Please enter password
Confirm Password
Please enter confirm password
Submit

DEPARTMENT *Computer Science & Information Technology*	NEPATHYA COLLEGE, JANAKINAGAR , BUTWAL	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : Custom Validation	
PAPER CODE: *CSE-314N*		***ISSUE DATE : 1ST Jan, 2018***
EXPERIMENT NO. : 08		***SEMESTER : 6TH Semester***
LABORATORY : *NET CENTRIC COMPUTING LAB*		***PAGE NO. : 23-24***

AIM : Write a program to implement the custom validation.

PROGRAM :

Student.cs

```
using MvcCoreCustomModelValidation_Demo.CustomValidation;
using System;
using System.ComponentModel.DataAnnotations;

namespace MvcCoreCustomModelValidation_Demo.Models
{
   public class Student
   {
      [Key]
      public int Id { get; set; }

      [Required(ErrorMessage = "Please enter name")]
      public string Name { get; set; }

      [Required(ErrorMessage = "Please choose admission date.")]
      [Display(Name = "Admission Date")]
      [DataType(DataType.Date)]
      [CustomAdmissionDate(ErrorMessage = "Admission Date must be less than or equal to
Today's Date.")]
      public DateTime AdmissionDate { get; set; }

      [Display(Name = "Date of Birth")]
      [DataType(DataType.Date)]
      [Min18Years]
      public DateTime DateofBirth { get; set; }
```

```
    }
}
```

Valiation.cs

```
using System;
using System.ComponentModel.DataAnnotations;

namespace MvcCoreCustomModelValidation_Demo.CustomValidation
{
    public class CustomAdmissionDate : ValidationAttribute
    {
        public override bool IsValid(object value)
        {
            DateTime dateTime = Convert.ToDateTime(value);
            return dateTime <= DateTime.Now;
        }
    }
}
```

Model.cs

```
using MvcCoreCustomModelValidation_Demo.Models;
using System;
using System.ComponentModel.DataAnnotations;

namespace MvcCoreCustomModelValidation_Demo.CustomValidation
{
    public class Min18Years : ValidationAttribute
    {
        protected override ValidationResult IsValid(object value, ValidationContext
validationContext)
        {
            var student = (Student)validationContext.ObjectInstance;

            if (student.DateofBirth == null)
                return new ValidationResult("Date of Birth is required.");

            var age = DateTime.Today.Year - student.DateofBirth.Year;

            return (age >= 18)
                ? ValidationResult.Success
                : new ValidationResult("Student should be at least 18 years old.");
        }
    }
}
```

HomeController.cs

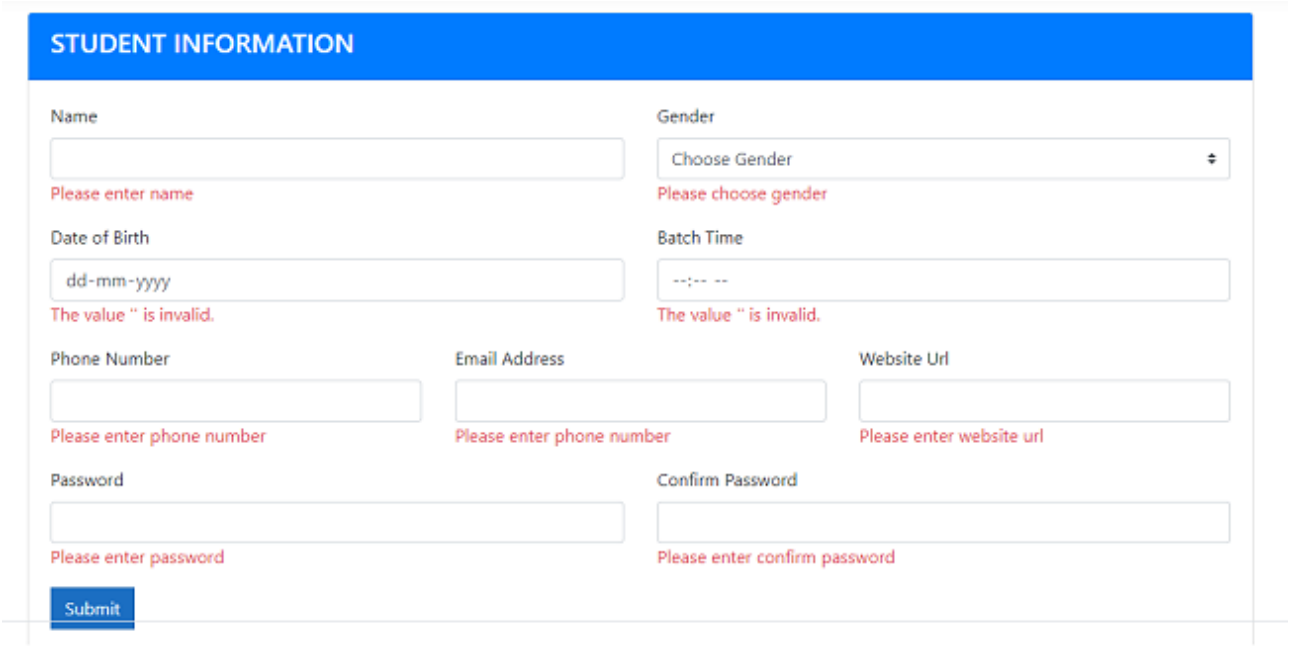
```
using Microsoft.AspNetCore.Mvc;
using MvcCoreCustomModelValidation_Demo.Models;

namespace MvcCoreCustomModelValidation_Demo.Controllers
{
   public class HomeController : Controller
   {
      public IActionResult Index()
      {
         return View();
      }

      public IActionResult New()
      {
         return View();
      }

      [HttpPost]
      [ValidateAntiForgeryToken]
      public IActionResult New(Student student)
      {
         if (ModelState.IsValid)
         {
            RedirectToAction("Index");
         }
         return View();
      }
   }
}
```

Index.cshtml

```
@model MvcCoreCustomModelValidation_Demo.Models.Student

@{
   ViewData["Title"] = "New";
}

<div class="card">
   <div class="card-header">
      <h4 class="text-uppercase">Student Information</h4>
   </div>
   <div class="card-body">
```

```
    <form asp-action="New">
      <div class="form-group">
        <label asp-for="Name" class="label-control"></label>
        <input asp-for="Name" class="form-control" />
        <span asp-validation-for="Name" class="text-danger"></span>
      </div>
      <div class="row">
        <div class="col-md-6">
          <div class="form-group">
            <label asp-for="AdmissionDate" class="label-control"></label>
            <input asp-for="AdmissionDate" class="form-control" />
            <span asp-validation-for="AdmissionDate" class="text-danger"></span>
          </div>
        </div>
        <div class="col-md-6">
          <div class="form-group">
            <label asp-for="DateofBirth" class="label-control"></label>
            <input asp-for="DateofBirth" class="form-control" />
            <span asp-validation-for="DateofBirth" class="text-danger"></span>
          </div>
        </div>
      </div>
      <div class="form-group">
        <button type="submit" class="btn btn-sm btn-primary rounded-0">Submit</button>
      </div>
    </form>
  </div>
</div>
```

OUTPUT:


STUDENT INFORMATION
Name
Please enter name
Gender
Choose Gender
Please choose gender
Date of Birth
dd-mm-yyyy
The value '' is invalid.
Batch Time
--:-- --
The value '' is invalid.
Phone Number
Please enter phone number
Email Address
Please enter phone number
Website Url
Please enter website url
Password
Please enter password
Confirm Password
Please enter confirm password
Submit

DEPARTMENT *Computer Science & Information Technology*	NEPATHYA COLLEGE, JANAKINAGAR , BUTWAL	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : Exception Handling .	
PAPER CODE: *CSE-314N*		***ISSUE DATE : 1ST Jan, 2018***
EXPERIMENT NO. : *09*		***SEMESTER :*** *6TH Semester*
LABORATORY : *NET CENTRIC COMPUTING LAB*		***PAGE NO. :*** *25-26*

AIM : Write a Program to illustrate Exception Handling in C#

PROGRAM :

```
using System;
class MyClient
{
   public static void Main()
   {
      int x = 0;
      int div = 0;
      try
      {
         div = 100 / x;
         Console.WriteLine("This linein not executed");
      }
      catch (DivideByZeroException)
      {
         Console.WriteLine("Exception occured");
      }
      Console.WriteLine($"Result is {div}");
   }
}
```

OUTPUT:

Exception Occurred

Result is 0

DEPARTMENT *Computer Science & Information Technology*	NEPATHYA COLLEGE, JANAKINAGAR , BUTWAL	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE : Custom Exception	
PAPER CODE: *CSE-314N*		***ISSUE DATE :*** *1ST Jan, 2018*
EXPERIMENT NO. : *10*		***SEMESTER :*** *6TH Semester*
LABORATORY : *NET CENTRIC COMPUTING LAB*		***PAGE NO. :*** *27-30*

AIM : Write a c# program to implement custom exception

PROGRAM :

```
class Student
{
   public int StudentID { get; set; }
   public string StudentName { get; set; }
}

[Serializable]
class InvalidStudentNameException : Exception
{
   public InvalidStudentNameException() {  }

   public InvalidStudentNameException(string name)
      : base(String.Format("Invalid Student Name: {0}", name))
   {

   }
}

class Program
{
   static void Main(string[] args)
   {
      Student newStudent = null;

      try
      {
         newStudent = new Student();
         newStudent.StudentName = "James000";
```

```
            ValidateStudent(newStudent);
        }
        catch(InvalidStudentNameException ex)
        {
            Console.WriteLine(ex.Message );
        }


        Console.ReadKey();
    }

    private static void ValidateStudent(Student std)
    {
        Regex regex = new Regex("^[a-zA-Z]+$");

        if (!regex.IsMatch(std.StudentName))
            throw new InvalidStudentNameException(std.StudentName);

    }
}
```

OUTPUT:

INVALID STUDENT NAME: James000