1. Create an XML Web Service to implement calculator with web methods to add, sub, multiply, divide which accepts decimal values. Consume this service through a web client application.

CalculatorWebService:

CalculatorService.asmx:

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

namespace CalculatorWebService
{
   [WebService(Namespace = "http://tempuri.org/")]
   [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
   [System.ComponentModel.ToolboxItem(false)]

   public class CalculatorService : System.Web.Services.WebService
   {
      [WebMethod]
      public decimal Add(decimal num1, decimal num2)
      {
         return num1 + num2;
      }
      [WebMethod]
      public decimal Subtract(decimal num1, decimal num2)
      {
         return num1 - num2;
      }
      [WebMethod]
      public decimal Multiply(decimal num1, decimal num2)
      {
         return num1 * num2;
      }
      [WebMethod]
      public decimal Divide(decimal num1, decimal num2)
      {
         if (num2 == 0)
         {
            throw new DivideByZeroException("Cannot divide by zero");
         }
         return num1 / num2;
      }
   }
}
```

CalculatorWebClient:

Default.aspx:

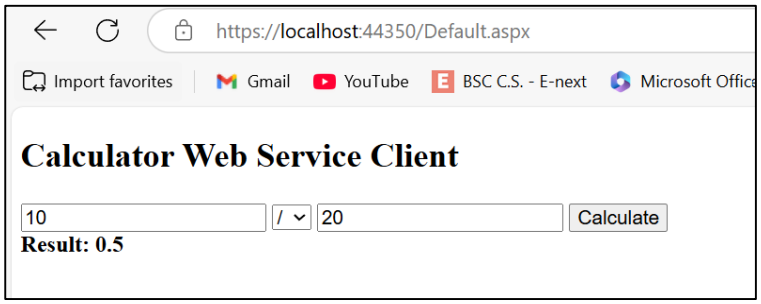
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="CalculatorWebClient.Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <h2>Calculator Web Service Client</h2>
         <asp:TextBox ID="txtNum1" runat="server" placeholder="First number" />
         <asp:DropDownList ID="ddlOperation" runat="server">
            <asp:ListItem Text="+" Value="Add" />
            <asp:ListItem Text="-" Value="Subtract" />
            <asp:ListItem Text="*" Value="Multiply" />
            <asp:ListItem Text="/" Value="Divide" />
         </asp:DropDownList>
         <asp:TextBox ID="txtNum2" runat="server" placeholder="Second number" />
         <asp:Button ID="btnCalculate" runat="server" Text="Calculate"
OnClick="btnCalculate_Click" />
         <br />
         <asp:Label ID="lblResult" runat="server" Font-Bold="true" />
         <asp:Label ID="lblError" runat="server" ForeColor="Red" />
      </div>
   </form>
</body>
</html>
```

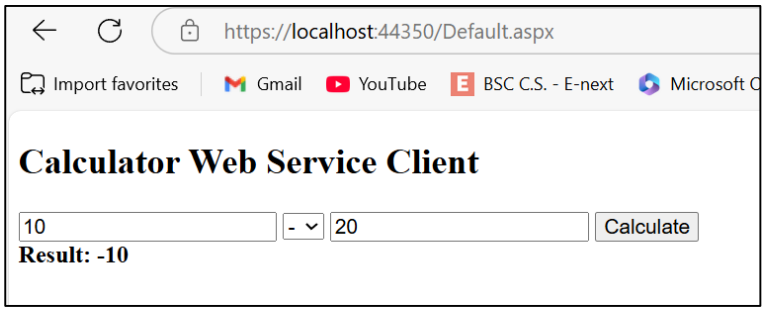
Default.aspx.cs

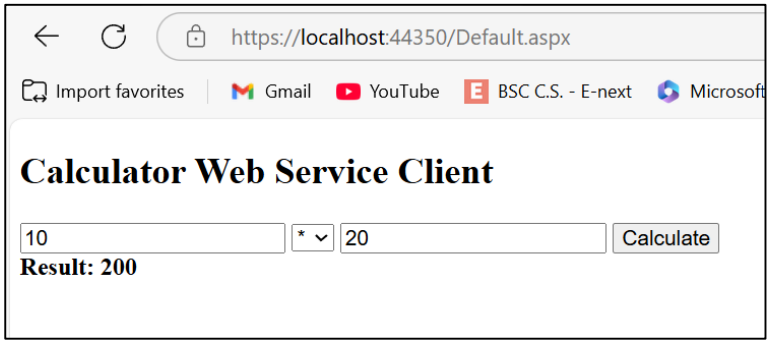
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace CalculatorWebClient
{
   public partial class Default : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         lblError.Text = "";
      }
```

```
    protected void btnCalculate_Click(object sender, EventArgs e)
    {
        try
        {
            decimal num1 = decimal.Parse(txtNum1.Text);
            decimal num2 = decimal.Parse(txtNum2.Text);
            string operation = ddlOperation.SelectedValue;
            localhost.CalculatorService calculator = new localhost.CalculatorService();
            decimal result = 0;
            switch (operation)
            {
                case "Add":
                    result = calculator.Add(num1, num2);
                    break;
                case "Subtract":
                    result = calculator.Subtract(num1, num2);
                    break;
                case "Multiply":
                    result = calculator.Multiply(num1, num2);
                    break;
                case "Divide":
                    result = calculator.Divide(num1, num2);
                    break;
            }
            lblResult.Text = $"Result: {result}";
            lblError.Text = "";
        }
        catch (FormatException)
        {
            lblError.Text = "Please enter valid decimal numbers";
            lblResult.Text = "";
        }
        catch (DivideByZeroException)
        {
            lblError.Text = "Cannot divide by zero";
            lblResult.Text = "";
        }
        catch (Exception ex)
        {
            lblError.Text = $"Error: {ex.Message}";
            lblResult.Text = "";
        }
    }
  }
}
```

Output:







2. Create an XML Web Service that performs show all records, add, update, delete and search operations on emp_info Database table. Design a Web client that consumes this service. (emp_info(id, name, department))

EmployeeWebService
Employee.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
namespace EmployeeWebService
{
   public class Employee
```

```
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Department { get; set; }
    }
}
```

EmployeeService.asmx.cs

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.Services;
namespace EmployeeWebService
{
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    public class EmployeeService : System.Web.Services.WebService
    {
        private string connectionString =
ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
        [WebMethod]
        public List<Employee> GetAllEmployees()
        {
            List<Employee> employees = new List<Employee>();
            using (SqlConnection con = new SqlConnection(connectionString))
            {
                SqlCommand cmd = new SqlCommand("SELECT id, name, department FROM
emp_info", con);
                con.Open();
                SqlDataReader rdr = cmd.ExecuteReader();
                while (rdr.Read())
                {
                    employees.Add(new Employee
                    {
                        Id = Convert.ToInt32(rdr["id"]),
                        Name = rdr["name"].ToString(),
                        Department = rdr["department"].ToString()
                    });
                }
            }
            return employees;
        }
```

```
    [WebMethod]
    public int AddEmployee(string name, string department)
    {
        using (SqlConnection con = new SqlConnection(connectionString))
        {
            SqlCommand cmd = new SqlCommand(
                "INSERT INTO emp_info (name, department) VALUES (@name,
@department); SELECT SCOPE_IDENTITY();",
                con);
            cmd.Parameters.AddWithValue("@name", name);
            cmd.Parameters.AddWithValue("@department", department);
            con.Open();
            return Convert.ToInt32(cmd.ExecuteScalar());
        }
    }
    [WebMethod]
    public bool UpdateEmployee(int id, string name, string department)
    {
        using (SqlConnection con = new SqlConnection(connectionString))
        {
            SqlCommand cmd = new SqlCommand(
                "UPDATE emp_info SET name = @name, department = @department WHERE
id = @id",
                con);
            cmd.Parameters.AddWithValue("@id", id);
            cmd.Parameters.AddWithValue("@name", name);
            cmd.Parameters.AddWithValue("@department", department);
            con.Open();
            return cmd.ExecuteNonQuery() > 0;
        }
    }
    [WebMethod]
    public bool DeleteEmployee(int id)
    {
        using (SqlConnection con = new SqlConnection(connectionString))
        {
            SqlCommand cmd = new SqlCommand("DELETE FROM emp_info WHERE id =
@id", con);
            cmd.Parameters.AddWithValue("@id", id);

            con.Open();
            return cmd.ExecuteNonQuery() > 0;
        }
    }
    [WebMethod]
    public List<Employee> SearchEmployees(string searchTerm)
```

```
        {
            List<Employee> employees = new List<Employee>();
            using (SqlConnection con = new SqlConnection(connectionString))
            {
                SqlCommand cmd = new SqlCommand(
                    "SELECT id, name, department FROM emp_info WHERE name LIKE @search
OR department LIKE @search",
                    con);
                cmd.Parameters.AddWithValue("@search", "%" + searchTerm + "%");
                con.Open();
                SqlDataReader rdr = cmd.ExecuteReader();
                while (rdr.Read())
                {
                    employees.Add(new Employee
                    {
                        Id = Convert.ToInt32(rdr["id"]),
                        Name = rdr["name"].ToString(),
                        Department = rdr["department"].ToString()
                    });
                }
            }
            return employees;
        }
    }
}
```

EmployeeWebClient
Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="EmployeeWebClient.Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
</head>
<body>
    <form id="form1" runat="server">
        <div class="container">
    <h2>Employee Management System</h2>
    <div class="panel panel-default">
        <div class="panel-heading">Search Employees</div>
        <div class="panel-body">
```

```
        <asp:TextBox ID="txtSearch" runat="server" CssClass="form-
control"></asp:TextBox>
        <asp:Button ID="btnSearch" runat="server" Text="Search"
          OnClick="btnSearch_Click" CssClass="btn btn-primary" />
        <asp:Button ID="btnShowAll" runat="server" Text="Show All"
          OnClick="btnShowAll_Click" CssClass="btn btn-default" />
      </div>
    </div>
    <asp:GridView ID="gvEmployees" runat="server" AutoGenerateColumns="False"
      DataKeyNames="Id" CssClass="table table-striped"
      OnRowEditing="gvEmployees_RowEditing"
      OnRowUpdating="gvEmployees_RowUpdating"
      OnRowCancelingEdit="gvEmployees_RowCancelingEdit"
      OnRowDeleting="gvEmployees_RowDeleting">
      <Columns>
        <asp:BoundField DataField="Id" HeaderText="ID" ReadOnly="true" />
        <asp:TemplateField HeaderText="Name">
          <ItemTemplate><%# Eval("Name") %></ItemTemplate>
          <EditItemTemplate>
            <asp:TextBox ID="txtName" runat="server" Text='<%# Bind("Name")
%>'></asp:TextBox>
          </EditItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Department">
          <ItemTemplate><%# Eval("Department") %></ItemTemplate>
          <EditItemTemplate>
            <asp:TextBox ID="txtDept" runat="server" Text='<%# Bind("Department")
%>'></asp:TextBox>
          </EditItemTemplate>
        </asp:TemplateField>
        <asp:CommandField ShowEditButton="true" ShowDeleteButton="true" />
      </Columns>
    </asp:GridView>
    <div class="panel panel-default">
      <div class="panel-heading">Add New Employee</div>
      <div class="panel-body">
        <div class="form-group">
          <label>Name:</label>
          <asp:TextBox ID="txtNewName" runat="server" CssClass="form-
control"></asp:TextBox>
        </div>
        <div class="form-group">
          <label>Department:</label>
          <asp:TextBox ID="txtNewDept" runat="server" CssClass="form-
control"></asp:TextBox>
        </div>
```

```
            <asp:Button ID="btnAdd" runat="server" Text="Add Employee"
                OnClick="btnAdd_Click" CssClass="btn btn-success" />
        </div>
    </div>
    <asp:Label ID="lblMessage" runat="server" CssClass="alert"
Visible="false"></asp:Label>
</div>
    </form>
</body>
</html>
```

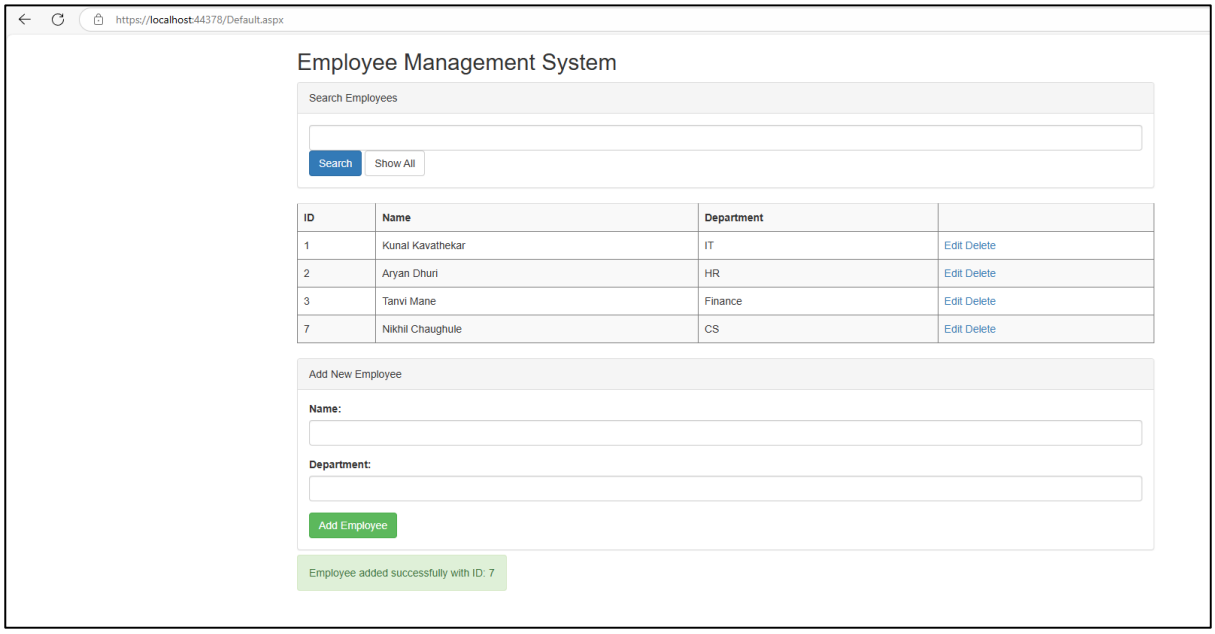
Default.aspx.cs

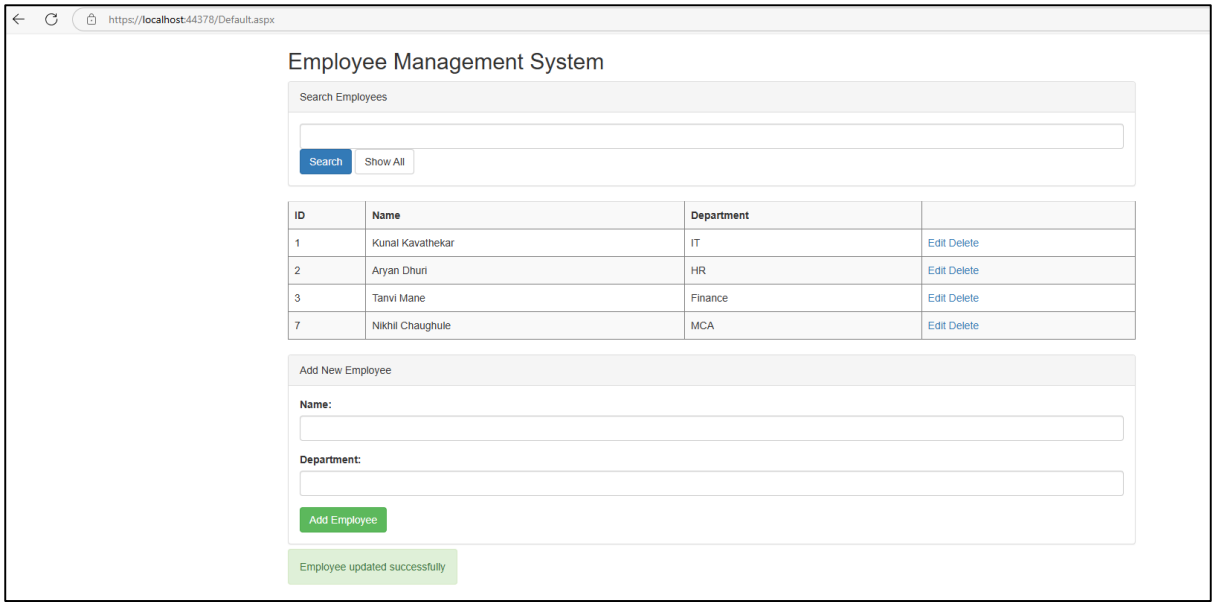
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace EmployeeWebClient
{
    public partial class Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                BindEmployees();
            }
        }
        private void BindEmployees()
        {
            var service = new localhost.EmployeeService();
            gvEmployees.DataSource = service.GetAllEmployees().ToList();
            gvEmployees.DataBind();
        }
        protected void btnSearch_Click(object sender, EventArgs e)
        {
            var service = new localhost.EmployeeService();
            gvEmployees.DataSource = service.SearchEmployees(txtSearch.Text).ToList();
            gvEmployees.DataBind();
        }
        protected void btnShowAll_Click(object sender, EventArgs e)
        {
            txtSearch.Text = "";
            BindEmployees();
        }
```

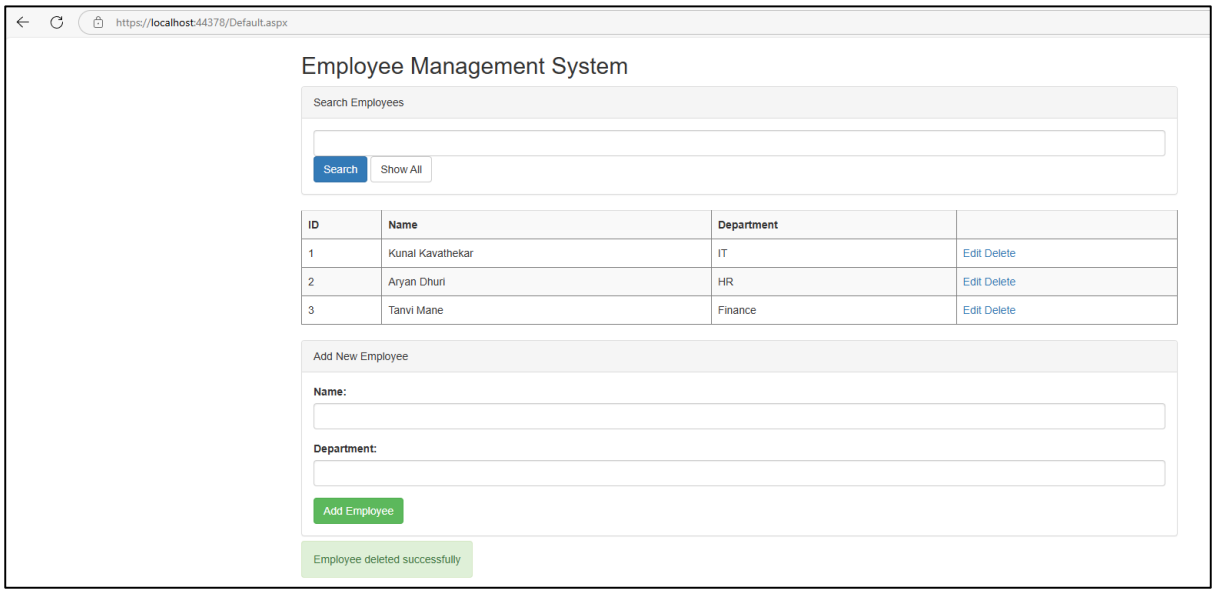
```
    protected void btnAdd_Click(object sender, EventArgs e)
    {
        var service = new localhost.EmployeeService();
        int newId = service.AddEmployee(txtNewName.Text, txtNewDept.Text);
        if (newId > 0)
        {
            ShowMessage($"Employee added successfully with ID: {newId}", "success");
            txtNewName.Text = "";
            txtNewDept.Text = "";
            BindEmployees();
        }
        else
        {
            ShowMessage("Failed to add employee", "danger");
        }
    }
    protected void gvEmployees_RowEditing(object sender, GridViewEditEventArgs e)
    {
        gvEmployees.EditIndex = e.NewEditIndex;
        BindEmployees();
    }
    protected void gvEmployees_RowUpdating(object sender, GridViewUpdateEventArgs
e)
    {
        int id = Convert.ToInt32(gvEmployees.DataKeys[e.RowIndex].Value);
        string name =
((TextBox)gvEmployees.Rows[e.RowIndex].FindControl("txtName")).Text;
        string dept =
((TextBox)gvEmployees.Rows[e.RowIndex].FindControl("txtDept")).Text;
        var service = new localhost.EmployeeService();
        bool success = service.UpdateEmployee(id, name, dept);
        if (success)
        {
            ShowMessage("Employee updated successfully", "success");
            gvEmployees.EditIndex = -1;
            BindEmployees();
        }
        else
        {
            ShowMessage("Failed to update employee", "danger");
        }
    }
    protected void gvEmployees_RowCancelingEdit(object sender,
GridViewCancelEditEventArgs e)
    {
        gvEmployees.EditIndex = -1;
```

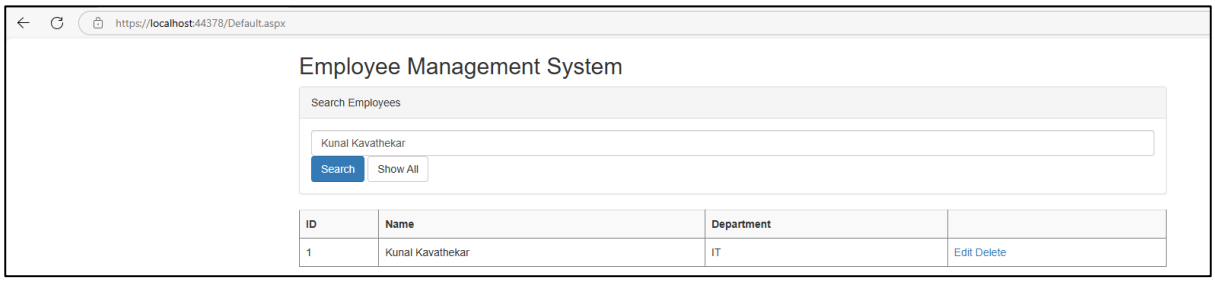
```
            BindEmployees();
        }
        protected void gvEmployees_RowDeleting(object sender, GridViewDeleteEventArgs e)
        {
            int id = Convert.ToInt32(gvEmployees.DataKeys[e.RowIndex].Value);
            var service = new localhost.EmployeeService();
            bool success = service.DeleteEmployee(id);
            if (success)
            {
                ShowMessage("Employee deleted successfully", "success");
                BindEmployees();
            }
            else
            {
                ShowMessage("Failed to delete employee", "danger");
            }
        }
        private void ShowMessage(string message, string type)
        {
            lblMessage.Text = message;
            lblMessage.CssClass = $"alert alert-{type}";
            lblMessage.Visible = true;
        }
    }
}
```

Output:









3. Design a Web Service using WCF for simple Calculator and consume it with client application.

WcfCalculatorService
ICalculatorService.cs

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

```
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;
namespace WcfCalculatorService
{
   [ServiceContract]
   public interface ICalculatorService
   {
      [OperationContract]
      decimal Add(decimal num1, decimal num2);

      [OperationContract]
      decimal Subtract(decimal num1, decimal num2);

      [OperationContract]
      decimal Multiply(decimal num1, decimal num2);

      [OperationContract]
      decimal Divide(decimal num1, decimal num2);
   }
}
```

CalculatorService.svc.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;
namespace WcfCalculatorService
{
CalculatorService.svc or CalculatorService.svc.cs at the Solution Explorer and start
debugging.
   public class CalculatorService : ICalculatorService
   {
      public decimal Add(decimal num1, decimal num2)
      {
         return num1 + num2;
      }
      public decimal Subtract(decimal num1, decimal num2)
      {
         return num1 - num2;
      }
      public decimal Multiply(decimal num1, decimal num2)
      {
         return num1 * num2;
```

```
        }
        public decimal Divide(decimal num1, decimal num2)
        {
            if (num2 == 0)
            {
                throw new FaultException("Cannot divide by zero");
            }
            return num1 / num2;
        }
    }
}
```

WcfCalculatorService:

Default.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="WcfCalculatorClient.Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>WCF Calculator Client</h2>
            <asp:TextBox ID="txtNum1" runat="server" placeholder="First number" />
            <asp:DropDownList ID="ddlOperation" runat="server">
                <asp:ListItem Text="+" Value="Add" />
                <asp:ListItem Text="-" Value="Subtract" />
                <asp:ListItem Text="*" Value="Multiply" />
                <asp:ListItem Text="/" Value="Divide" />
            </asp:DropDownList>
            <asp:TextBox ID="txtNum2" runat="server" placeholder="Second number" />
            <asp:Button ID="btnCalculate" runat="server" Text="Calculate"
OnClick="btnCalculate_Click" />
            <br />
            <asp:Label ID="lblResult" runat="server" Font-Bold="true" />
            <asp:Label ID="lblError" runat="server" ForeColor="Red" />
        </div>
    </form>
</body>
</html>
```

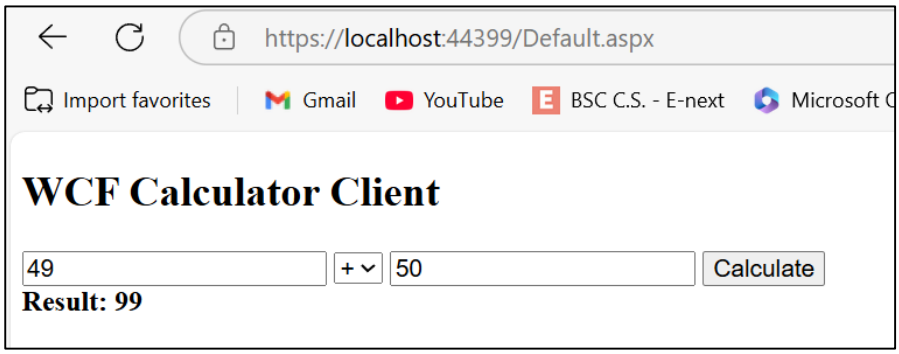
Default.aspx.cs

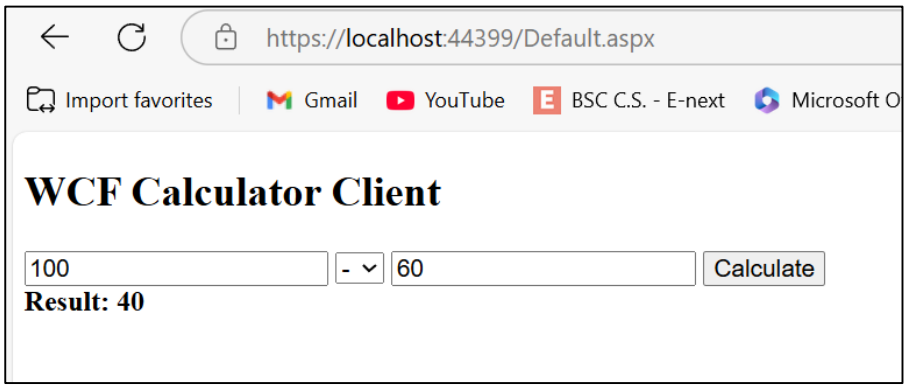
```
using System;
using System.Collections.Generic;
```

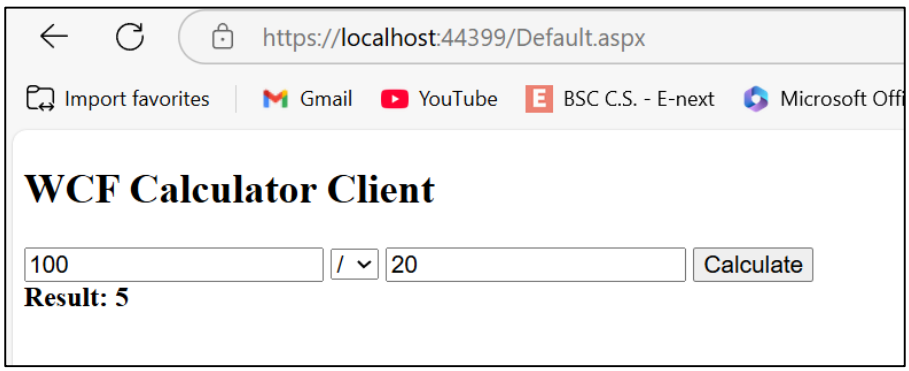
```
using System.Linq;
using System.ServiceModel;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace WcfCalculatorClient
{
   public partial class Default : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         lblError.Text = "";
      }
      protected void btnCalculate_Click(object sender, EventArgs e)
      {
         try
      {
         decimal num1 = decimal.Parse(txtNum1.Text);
         decimal num2 = decimal.Parse(txtNum2.Text);
         string operation = ddlOperation.SelectedValue;
         CalculatorServiceRef.CalculatorServiceClient client = new
CalculatorServiceRef.CalculatorServiceClient();
         decimal result = 0;
         switch (operation)
         {
            case "Add":
               result = client.Add(num1, num2);
               break;
            case "Subtract":
               result = client.Subtract(num1, num2);
               break;
            case "Multiply":
               result = client.Multiply(num1, num2);
               break;
            case "Divide":
               result = client.Divide(num1, num2);
               break;
         }
         lblResult.Text = $"Result: {result}";
         lblError.Text = "";
         client.Close();
      }
      catch (FormatException)
      {
         lblError.Text = "Please enter valid decimal numbers";
         lblResult.Text = "";
```

```
        }
        catch (FaultException fe)
        {
            lblError.Text = $"Service error: {fe.Message}";
            lblResult.Text = "";
        }
        catch (Exception ex)
        {
            lblError.Text = $"Error: {ex.Message}";
            lblResult.Text = "";
        }
        }
    }
}
```

Output:







4. Create a WCF Web Service that performs show all records, add, update, delete and search operations on product_info Database table. Design a Web client that consumes this service. (product_info(pid, name, salary))

ProductWcfService
Product.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.Web;
namespace ProductWcfService
{
   [DataContract]
   public class Product
   {
      [DataMember]
      public int Pid { get; set; }

      [DataMember]
      public string Name { get; set; }

      [DataMember]
      public decimal Salary { get; set; }
   }
}
```

IproductService.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;
namespace ProductWcfService
{
```

```
    [ServiceContract]
    public interface IProductService
    {
        [OperationContract]
        List<Product> GetAllProducts();

        [OperationContract]
        int AddProduct(string name, decimal salary);

        [OperationContract]
        bool UpdateProduct(int pid, string name, decimal salary);

        [OperationContract]
        bool DeleteProduct(int pid);

        [OperationContract]
        List<Product> SearchProducts(string searchTerm);
    }
}
```

ProductService.svc.cs

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data.SqlClient;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;
namespace ProductWcfService
{
    public class ProductService : IProductService
    {
        private string connectionString =
ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
        public List<Product> GetAllProducts()
        {
            var products = new List<Product>();
            using (var con = new SqlConnection(connectionString))
            using (var cmd = new SqlCommand("SELECT pid, name, salary FROM
product_info", con))
            {
                con.Open();
                using (var rdr = cmd.ExecuteReader())
                {
                    while (rdr.Read())
```

```
            {
                products.Add(new Product
                {
                    Pid = Convert.ToInt32(rdr["pid"]),
                    Name = rdr["name"].ToString(),
                    Salary = Convert.ToDecimal(rdr["salary"])
                });
            }
        }
    }
    return products;
}
public int AddProduct(string name, decimal salary)
{
    using (var con = new SqlConnection(connectionString))
    using (var cmd = new SqlCommand(
        "INSERT INTO product_info (name, salary) VALUES (@name, @salary);
SELECT SCOPE_IDENTITY();",
        con))
    {
        cmd.Parameters.AddWithValue("@name", name);
        cmd.Parameters.AddWithValue("@salary", salary);
        con.Open();
        return Convert.ToInt32(cmd.ExecuteScalar());
    }
}
public bool UpdateProduct(int pid, string name, decimal salary)
{
    using (var con = new SqlConnection(connectionString))
    using (var cmd = new SqlCommand(
        "UPDATE product_info SET name = @name, salary = @salary WHERE pid =
@pid",
        con))
    {
        cmd.Parameters.AddWithValue("@pid", pid);
        cmd.Parameters.AddWithValue("@name", name);
        cmd.Parameters.AddWithValue("@salary", salary);

        con.Open();
        return cmd.ExecuteNonQuery() > 0;
    }
}
public bool DeleteProduct(int pid)
{
    using (var con = new SqlConnection(connectionString))
```

```
        using (var cmd = new SqlCommand("DELETE FROM product_info WHERE pid =
@pid", con))
        {
            cmd.Parameters.AddWithValue("@pid", pid);
            con.Open();
            return cmd.ExecuteNonQuery() > 0;
        }
    }
    public List<Product> SearchProducts(string searchTerm)
    {
        var products = new List<Product>();
        if (string.IsNullOrWhiteSpace(searchTerm))
        {
            return GetAllProducts();
        }
        using (var con = new SqlConnection(connectionString))
        using (var cmd = new SqlCommand(
            "SELECT pid, name, salary FROM product_info WHERE name LIKE @search",
            con))
        {
            cmd.Parameters.AddWithValue("@search", "%" + searchTerm.Trim() + "%");
            con.Open();
            using (var rdr = cmd.ExecuteReader())
            {
                while (rdr.Read())
                {
                    products.Add(new Product
                    {
                        Pid = Convert.ToInt32(rdr["pid"]),
                        Name = rdr["name"].ToString(),
                        Salary = Convert.ToDecimal(rdr["salary"])
                    });
                }
            }
        }
        return products;
    }
  }
}
```

ProductWcfClient
Default.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="ProductWcfClient.Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
    <title></title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"/>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</head>
<body>
    <form id="form1" runat="server">
        <div class="container">
            <h2>Product Management System</h2>
            <div class="panel panel-default">
                <div class="panel-heading">Search Products</div>
                <div class="panel-body">
                    <asp:TextBox ID="txtSearch" runat="server" CssClass="form-
control"></asp:TextBox>
                    <asp:Button ID="btnSearch" runat="server" Text="Search"
                        OnClick="btnSearch_Click" CssClass="btn btn-primary" />
                    <asp:Button ID="btnShowAll" runat="server" Text="Show All"
                        OnClick="btnShowAll_Click" CssClass="btn btn-default" />
                </div>
            </div>
            <asp:GridView ID="gvProducts" runat="server" AutoGenerateColumns="False"
                DataKeyNames="Pid" CssClass="table table-striped"
                OnRowEditing="gvProducts_RowEditing"
                OnRowUpdating="gvProducts_RowUpdating"
                OnRowCancelingEdit="gvProducts_RowCancelingEdit"
                OnRowDeleting="gvProducts_RowDeleting">
                <Columns>
                    <asp:BoundField DataField="Pid" HeaderText="ID" ReadOnly="true" />
                    <asp:TemplateField HeaderText="Name">
                        <ItemTemplate><%# Eval("Name") %></ItemTemplate>
                        <EditItemTemplate>
                            <asp:TextBox ID="txtName" runat="server" Text='<%# Bind("Name") %>'
CssClass="form-control"></asp:TextBox>
                        </EditItemTemplate>
                    </asp:TemplateField>
                    <asp:TemplateField HeaderText="Salary">
                        <ItemTemplate><%# Eval("Salary", "{0:C}") %></ItemTemplate>
                        <EditItemTemplate>
                            <asp:TextBox ID="txtSalary" runat="server" Text='<%# Bind("Salary")
%>' CssClass="form-control"></asp:TextBox>
                        </EditItemTemplate>
                    </asp:TemplateField>
                    <asp:CommandField ShowEditButton="true" ShowDeleteButton="true" />
```

```
            </Columns>
        </asp:GridView>
        <div class="panel panel-default">
            <div class="panel-heading">Add New Product</div>
            <div class="panel-body">
                <div class="form-group">
                    <label>Name:</label>
                    <asp:TextBox ID="txtNewName" runat="server" CssClass="form-
control"></asp:TextBox>
                </div>
                <div class="form-group">
                    <label>Salary:</label>
                    <asp:TextBox ID="txtNewSalary" runat="server" CssClass="form-
control"></asp:TextBox>
                </div>
                <asp:Button ID="btnAdd" runat="server" Text="Add Product"
                    OnClick="btnAdd_Click" CssClass="btn btn-success" />
            </div>
        </div>
        <asp:Label ID="lblMessage" runat="server" CssClass="alert"
Visible="false"></asp:Label>
    </div>
  </form>
</body>
</html>
```

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace ProductWcfClient
{
  public partial class Default : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
      if (!IsPostBack)
      {
        BindProducts();
      }
    }
    private void BindProducts()
    {
```

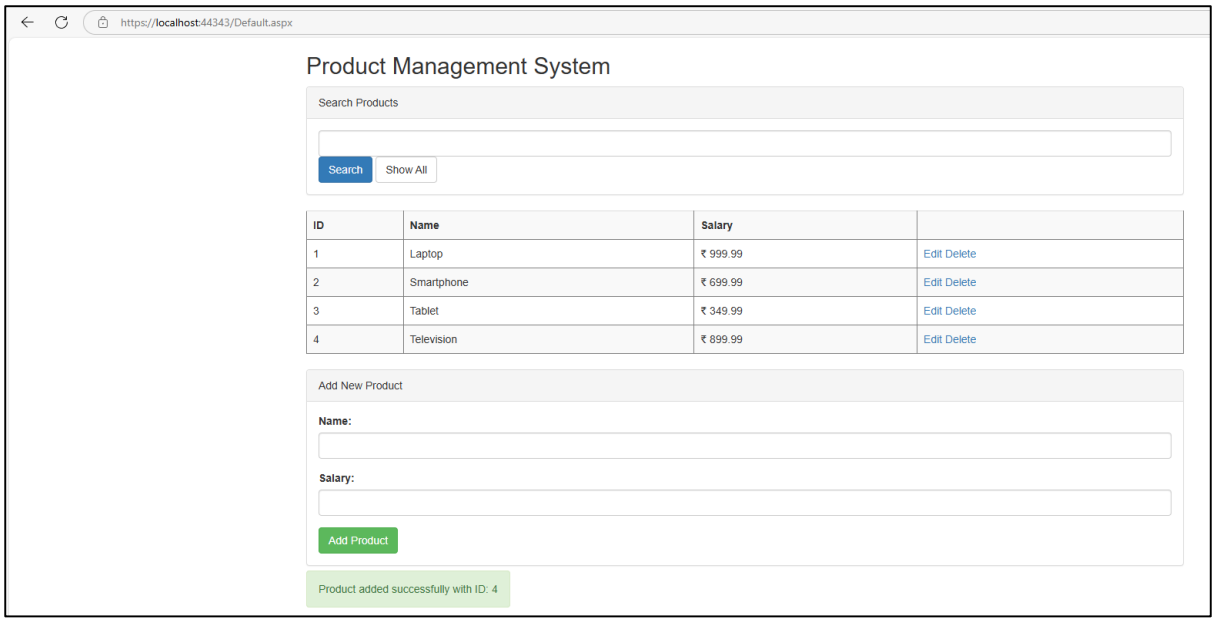
```
        using (var client = new ProductServiceRef.ProductServiceClient())
        {
            gvProducts.DataSource = client.GetAllProducts().ToList();
            gvProducts.DataBind();
        }
    }
    protected void btnSearch_Click(object sender, EventArgs e)
    {
        using (var client = new ProductServiceRef.ProductServiceClient())
        {
            try
            {
                var results = client.SearchProducts(txtSearch.Text.Trim());

                if (results != null && results.Count() > 0)  // Changed from results.Any()
                {
                    gvProducts.DataSource = results.ToList();
                    gvProducts.DataBind();
                    ShowMessage($"Found {results.Count()} matching products", "success");  //
Changed from results.Count
                }
                else
                {
                    gvProducts.DataSource = null;
                    gvProducts.DataBind();
                    ShowMessage("No products found matching your search", "info");
                }
            }
            catch (Exception ex)
            {
                ShowMessage("Search error: " + ex.Message, "danger");
            }
        }
    }
    protected void btnShowAll_Click(object sender, EventArgs e)
    {
        txtSearch.Text = "";
        BindProducts();
    }
    protected void btnAdd_Click(object sender, EventArgs e)
    {
        if (!decimal.TryParse(txtNewSalary.Text, out decimal salary))
        {
            ShowMessage("Please enter a valid salary", "danger");
            return;
        }
```

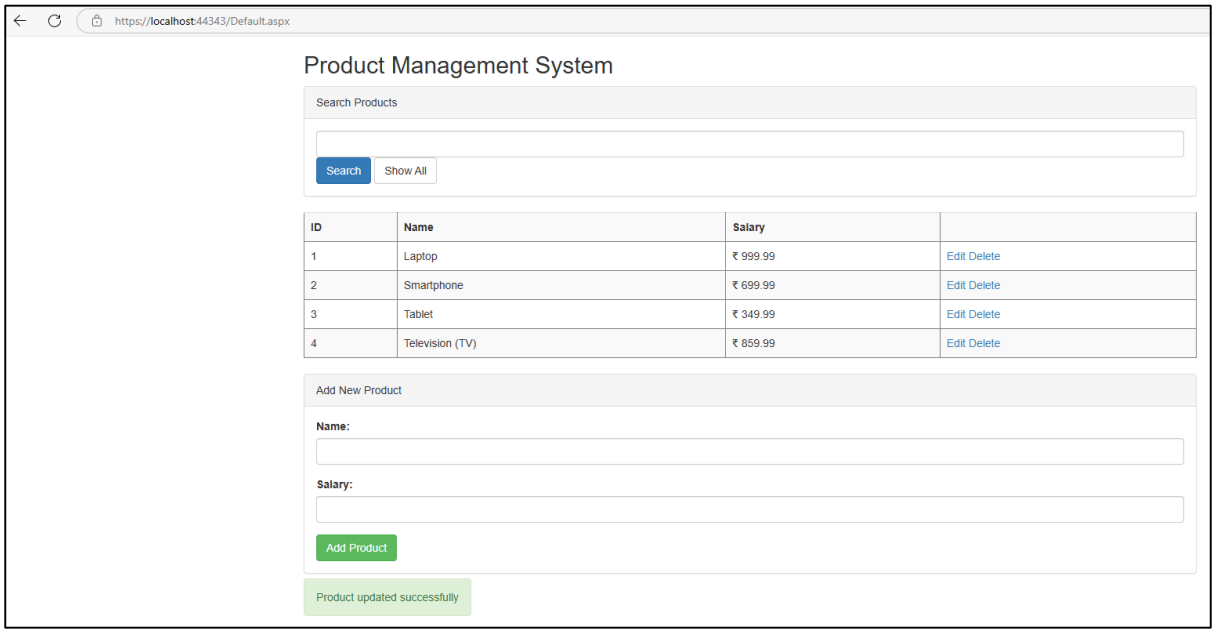
```
        using (var client = new ProductServiceRef.ProductServiceClient())
        {
            try
            {
                int newId = client.AddProduct(txtNewName.Text.Trim(), salary);
                if (newId > 0)
                {
                    ShowMessage($"Product added successfully with ID: {newId}", "success");
                    txtNewName.Text = "";
                    txtNewSalary.Text = "";
                    BindProducts();
                }
                else
                {
                    ShowMessage("Failed to add product", "danger");
                }
            }
            catch (Exception ex)
            {
                ShowMessage("Error adding product: " + ex.Message, "danger");
            }
        }
    }
    protected void gvProducts_RowEditing(object sender, GridViewEditEventArgs e)
    {
        gvProducts.EditIndex = e.NewEditIndex;
        BindProducts();
    }
    protected void gvProducts_RowUpdating(object sender, GridViewUpdateEventArgs e)
    {
        int pid = Convert.ToInt32(gvProducts.DataKeys[e.RowIndex].Value);
        string name =
((TextBox)gvProducts.Rows[e.RowIndex].FindControl("txtName")).Text;
        string salaryText =
((TextBox)gvProducts.Rows[e.RowIndex].FindControl("txtSalary")).Text;
        if (!decimal.TryParse(salaryText, out decimal salary))
        {
            ShowMessage("Please enter a valid salary", "danger");
            return;
        }
        using (var client = new ProductServiceRef.ProductServiceClient())
        {
            try
            {
                bool success = client.UpdateProduct(pid, name, salary);
                if (success)
```

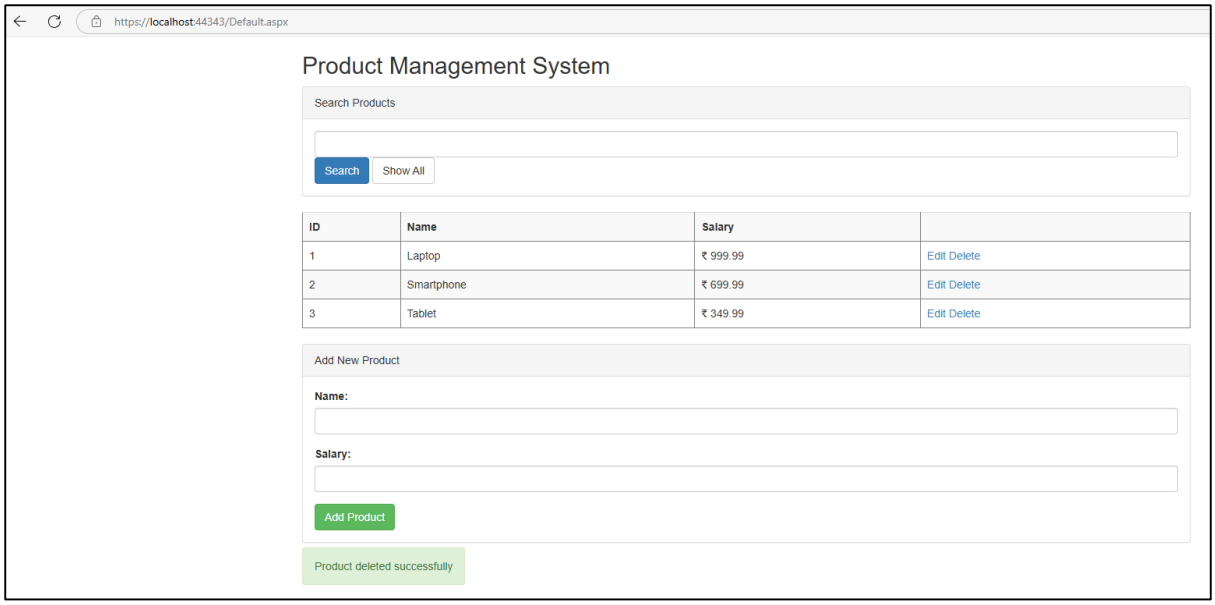
```
            {
                ShowMessage("Product updated successfully", "success");
                gvProducts.EditIndex = -1;
                BindProducts();
            }
            else
            {
                ShowMessage("Failed to update product", "danger");
            }
        }
        catch (Exception ex)
        {
            ShowMessage("Error updating product: " + ex.Message, "danger");
        }
    }
}
protected void gvProducts_RowCancelingEdit(object sender,
GridViewCancelEditEventArgs e)
{
    gvProducts.EditIndex = -1;
    BindProducts();
}
protected void gvProducts_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    int pid = Convert.ToInt32(gvProducts.DataKeys[e.RowIndex].Value);
    using (var client = new ProductServiceRef.ProductServiceClient())
    {
        try
        {
            bool success = client.DeleteProduct(pid);
            if (success)
            {
                ShowMessage("Product deleted successfully", "success");
                BindProducts();
            }
            else
            {
                ShowMessage("Failed to delete product", "danger");
            }
        }
        catch (Exception ex)
        {
            ShowMessage("Error deleting product: " + ex.Message, "danger");
        }
    }
}
```

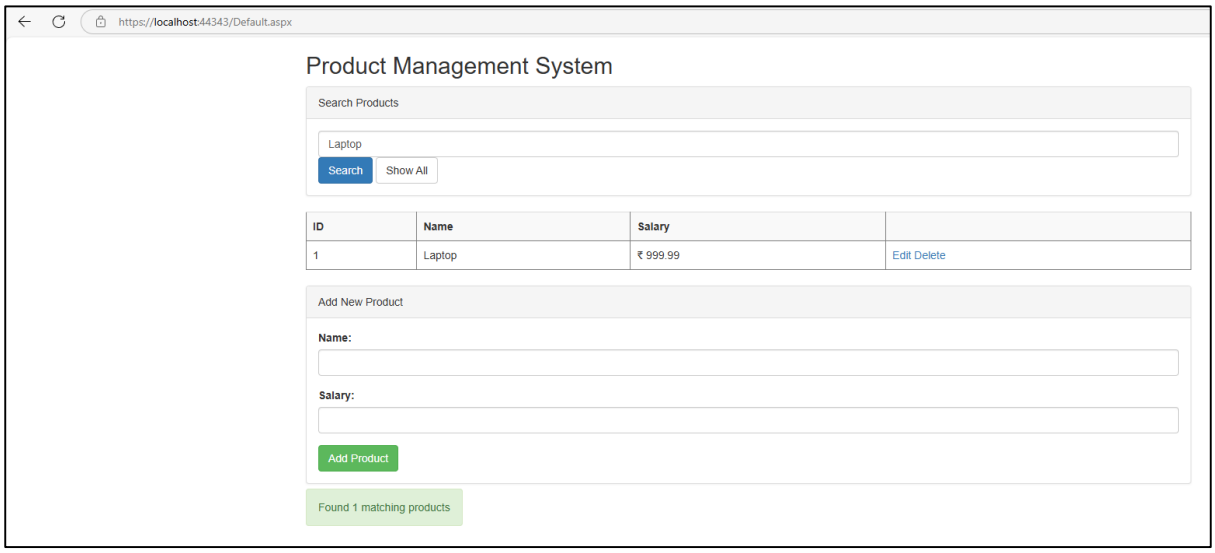
```
        private void ShowMessage(string message, string type)
        {
            lblMessage.Text = message;
            lblMessage.CssClass = $"alert alert-{type}";
            lblMessage.Visible = true;
        }
    }
}
```

Output:






https://localhost:44343/Default.aspx
Product Management System
Search Products
Search
Show All
ID
Name
Salary
1
Laptop
₹ 999.99
Edit Delete
2
Smartphone
₹ 699.99
Edit Delete
3
Tablet
₹ 349.99
Edit Delete
Add New Product
Name:
Salary:
Add Product
Product deleted successfully


https://localhost:44343/Default.aspx
Product Management System
Search Products
Laptop
Search
Show All
ID
Name
Salary
1
Laptop
₹ 999.99
Edit Delete
Add New Product
Name:
Salary:
Add Product
Found 1 matching products