ASP .NET MVC Assessment

Sasikumar S 05-04-2022

**Problem Statement:**

Develop an application using ASP.NET MVC as front end and SQL server as backend

with login authentication (Email, Password) and make the valid user to land on

Products dashboard to perform CRUD operations

**Tables to be created**

User Table (UserID, Email, Password)

Products (ProductID,Name,Company,Price,Qty,ExpiryDate)

**Models:**

**User**

using System.ComponentModel.DataAnnotations;

namespace AssessmentQuestion.Models

{

public class User

{

[Key]

public int UserID { get; set; }

[Required,MaxLength(30)]

public string Email { get; set; }

[Required,MaxLength(15)]

public string Password { get; set; }

}

}

**Product**

using System;

using System.ComponentModel.DataAnnotations;

namespace AssessmentQuestion.Models

{

public class Product

{

[Key]

public int ProductID { get; set; }

[Required,MaxLength(30)]

public string Name { get; set; }

[Required,MaxLength(30)]

public string Campany { get; set; }

[Required,Range(1,50000)]

public int Price { get; set; }

[Required,Range(0,2000)]

public int Qty { get; set; }

[Required]

public DateTime ExpiryDate{get;set;}

}

}

**AppDBContext**

using System.Data.Entity;

namespace AssessmentQuestion.Models

{

public class AppDBContext : DbContext

{

public AppDBContext() : base("ProductConStr")

{

}

public DbSet<User> Users { get; set; }

public DbSet<Product> Products { get; set; }

}

}

**WebConfig** **File**:

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

<!--

For more information on how to configure your ASP.NET application, please visit

https://go.microsoft.com/fwlink/?LinkId=301880

-->

<configuration>

<configSections>

<!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->

<section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />

</configSections>

<appSettings>

<add key="webpages:Version" value="3.0.0.0" />

<add key="webpages:Enabled" value="false" />

<add key="ClientValidationEnabled" value="true" />

<add key="UnobtrusiveJavaScriptEnabled" value="true" />

</appSettings>

<system.web>

<compilation debug="true" targetFramework="4.7.2" />

<httpRuntime targetFramework="4.7.2" />

</system.web>

<runtime>

<assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">

<dependentAssembly>

<assemblyIdentity name="Antlr3.Runtime" publicKeyToken="eb42632606e9261f" />

<bindingRedirect oldVersion="0.0.0.0-3.5.0.2" newVersion="3.5.0.2" />

</dependentAssembly>

<dependentAssembly>

<assemblyIdentity name="Newtonsoft.Json" publicKeyToken="30ad4fe6b2a6aeed" />

<bindingRedirect oldVersion="0.0.0.0-12.0.0.0" newVersion="12.0.0.0" />

</dependentAssembly>

<dependentAssembly>

<assemblyIdentity name="System.Web.Optimization" publicKeyToken="31bf3856ad364e35" />

<bindingRedirect oldVersion="1.0.0.0-1.1.0.0" newVersion="1.1.0.0" />

</dependentAssembly>

<dependentAssembly>

<assemblyIdentity name="WebGrease" publicKeyToken="31bf3856ad364e35" />

<bindingRedirect oldVersion="0.0.0.0-1.6.5135.21930" newVersion="1.6.5135.21930" />

</dependentAssembly>

<dependentAssembly>

<assemblyIdentity name="System.Web.Helpers" publicKeyToken="31bf3856ad364e35" />

<bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0" />

</dependentAssembly>

<dependentAssembly>

<assemblyIdentity name="System.Web.WebPages" publicKeyToken="31bf3856ad364e35" />

<bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0" />

</dependentAssembly>

<dependentAssembly>

<assemblyIdentity name="System.Web.Mvc" publicKeyToken="31bf3856ad364e35" />

<bindingRedirect oldVersion="1.0.0.0-5.2.7.0" newVersion="5.2.7.0" />

</dependentAssembly>

</assemblyBinding>

</runtime>

<system.codedom>

<compilers>

<compiler language="c#;cs;csharp" extension=".cs" type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider, Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" warningLevel="4" compilerOptions="/langversion:default /nowarn:1659;1699;1701" />

<compiler language="vb;vbs;visualbasic;vbscript" extension=".vb" type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider, Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" warningLevel="4" compilerOptions="/langversion:default /nowarn:41008 /define:\_MYTYPE=\&quot;Web\&quot; /optionInfer+" />

</compilers>

</system.codedom>

<entityFramework>

<providers>

<provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />

</providers>

</entityFramework>

<connectionStrings>

<add name="ProductConStr" connectionString="Data Source=SASIKUMAR\SQLEXPRESS;Initial Catalog=ProductDB;Integrated Security=True;" providerName="System.Data.SqlClient" />

</connectionStrings>

</configuration>

**ConfigurationFile**

namespace AssessmentQuestion.Migrations

{

using AssessmentQuestion.Models;

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Data.Entity.Migrations;

using System.Linq;

internal sealed class Configuration : DbMigrationsConfiguration<AssessmentQuestion.Models.AppDBContext>

{

public Configuration()

{

AutomaticMigrationsEnabled = false;

}

protected override void Seed(AssessmentQuestion.Models.AppDBContext context)

{

IList<User> UserList = new List<User>();

UserList.Add(new User()

{

UserID= 1,

Email = "sasi@gmail.com",

Password = "sasi123"

});

context.Users.AddRange(UserList);

base.Seed(context);

}

}

}

**HomeController:**

using AssessmentQuestion.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace AssessmentQuestion.Controllers

{

public class HomeController : Controller

{

private AppDBContext db = new AppDBContext();

public ActionResult Index()

{

return View();

}

public ActionResult Contact()

{

ViewBag.Message = "Your contact page.";

return View();

}

[HttpGet]

public ActionResult Login()

{

return View();

}

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Login(User user)

{

using (AppDBContext db = new AppDBContext())

{

var obj = db.Users.Where(a => a.Email.Equals(user.Email) && a.Password.Equals(user.Password)).FirstOrDefault();

if (obj != null)

{

Session["Email"] = obj.Email.ToString();

return RedirectToAction("DashBoard", "Products");

}

else

{

ViewBag.Message = "user not found for given Email and Password";

return View();

}

}

}

[HttpGet]

public ActionResult Logout()

{

Session.Clear();

return RedirectToAction("Login", "Home");

}

}

}

**ProductController:**

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

using AssessmentQuestion.Models;

namespace AssessmentQuestion.Controllers

{

public class ProductsController : Controller

{

private AppDBContext db = new AppDBContext();

// GET: Products

public ActionResult Index()

{

if (Session["Email"] != null)

{

return View(db.Products.ToList());

}

else

{

return RedirectToAction("Login", "Home");

}

}

// GET: Products/Details/5

public ActionResult Details(int? id)

{

if (Session["Email"] != null)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Product product = db.Products.Find(id);

if (product == null)

{

return HttpNotFound();

}

return View(product);

}

else

{

return RedirectToAction("Login", "Home");

}

}

// GET: Products/Create

public ActionResult Create()

{

if (Session["Email"] != null)

{

return View();

}

else

{

return RedirectToAction("Login", "Home");

}

}

// POST: Products/Create

// To protect from overposting attacks, enable the specific properties you want to bind to, for

// more details see https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "ProductID,Name,Campany,Price,Qty,ExpiryDate")] Product product)

{

if (Session["Email"] != null)

{

if (ModelState.IsValid)

{

db.Products.Add(product);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(product);

}

else

{

return RedirectToAction("Login", "Home");

}

}

// GET: Products/Edit/5

public ActionResult Edit(int? id)

{

if (Session["Email"] != null)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Product product = db.Products.Find(id);

if (product == null)

{

return HttpNotFound();

}

return View(product);

}

else

{

return RedirectToAction("Login", "Home");

}

}

// POST: Products/Edit/5

// To protect from overposting attacks, enable the specific properties you want to bind to, for

// more details see https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "ProductID,Name,Campany,Price,Qty,ExpiryDate")] Product product)

{

if (ModelState.IsValid)

{

db.Entry(product).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

return View(product);

}

// GET: Products/Delete/5

public ActionResult Delete(int? id)

{

if (Session["Email"] != null)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Product product = db.Products.Find(id);

if (product == null)

{

return HttpNotFound();

}

return View(product);

}

else

{

return RedirectToAction("Login", "Home");

}

}

// POST: Products/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Product product = db.Products.Find(id);

db.Products.Remove(product);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

public ActionResult DashBoard()

{

if (Session["Email"] != null)

{

return View();

}

else

{

return RedirectToAction("Login", "Home");

}

}

}

}

**LayOut:**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>@ViewBag.Title - My ASP.NET Application</title>

@Styles.Render("~/Content/css")

@Scripts.Render("~/bundles/modernizr")

</head>

<body>

<div class="navbar navbar-inverse navbar-fixed-top">

<div class="container">

<div class="navbar-header">

<button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

@Html.ActionLink("SK Products", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })

</div>

<div class="navbar-collapse collapse">

<ul class="nav navbar-nav">

<li>@Html.ActionLink("Home", "Index", "Home")</li>

<li>@Html.ActionLink("Contact", "Contact", "Home")</li>

@if (Session["Email"] == null)

{

<li style="background-color:greenyellow;color:black;font-weight:bold">@Html.ActionLink("Login", "Login", "Home")</li>

}

else

{

<li style="background-color:orangered;color:cornsilk;font-weight:bold">@Html.ActionLink("Logout", "Logout", "Home")</li>

<li style="color:wheat;margin-left:500px">@Session["Email"].ToString()</li>

}

</ul>

</div>

</div>

</div>

<div class="container body-content">

@RenderBody()

<hr />

<footer>

<p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>

</footer>

</div>

@Scripts.Render("~/bundles/jquery")

@Scripts.Render("~/bundles/bootstrap")

@RenderSection("scripts", required: false)

</body>

</html>

**Login:**

@model AssessmentQuestion.Models.User

@{

ViewBag.Title = "Login";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Login</h2>

@using (Html.BeginForm("Login", "Home", FormMethod.Post))

{

<fieldset class="col-md-6">

@Html.AntiForgeryToken()

@Html.ValidationSummary(true)

<table>

<tr>

<td>@Html.LabelFor(a => a.Email)</td>

<td>@Html.TextBoxFor(a => a.Email)</td>

<td>@Html.ValidationMessageFor(a => a.Email)</td>

</tr>

<tr><td>&nbsp;</td></tr>

<tr>

<td>@Html.LabelFor(a => a.Password)</td>

<td>@Html.PasswordFor(a => a.Password)</td>

<td>@Html.ValidationMessageFor(a => a.Password)</td>

</tr>

<tr><td>&nbsp;</td></tr>

<tr>

<td></td>

<td><input type="submit" value="Login" /></td>

<td></td>

</tr>

</table>

<br />

@if (@ViewBag.Message != null)

{

<div style="border: 1px solid red">

@ViewBag.Message

</div>

}

</fieldset>

}

**DashBoard:**

@{

ViewBag.Title = "DashBoard";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>DashBoard</h2>

<h3> Hello @Session["Email"].ToString() </h3>

<div style="margin-top:20px">

@Html.ActionLink("Upload Product", "Create", "Products", null, new { @class = "btn btn-success" })

</div>

<div style="margin-top:20px">

@Html.ActionLink("View All Products", "Index", "Products", null, new { @class = "btn btn-success" })

</div>

**Create:**

@model AssessmentQuestion.Models.Product

@{

ViewBag.Title = "Create";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Create</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>Product</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.Campany, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Campany, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Campany, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.Price, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Price, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Price, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.Qty, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Qty, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Qty, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.ExpiryDate, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.ExpiryDate, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.ExpiryDate, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<input type="submit" value="Create" class="btn btn-default" />

</div>

</div>

</div>

}

<div>

@Html.ActionLink("Back to List", "Index")

</div>

**Index:**

@model IEnumerable<AssessmentQuestion.Models.Product>

@{

ViewBag.Title = "Index";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Index</h2>

<p>

@Html.ActionLink("Create New", "Create")

</p>

<table class="table">

<tr>

<th>

@Html.DisplayNameFor(model => model.Name)

</th>

<th>

@Html.DisplayNameFor(model => model.Campany)

</th>

<th>

@Html.DisplayNameFor(model => model.Price)

</th>

<th>

@Html.DisplayNameFor(model => model.Qty)

</th>

<th>

@Html.DisplayNameFor(model => model.ExpiryDate)

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.Name)

</td>

<td>

@Html.DisplayFor(modelItem => item.Campany)

</td>

<td>

@Html.DisplayFor(modelItem => item.Price)

</td>

<td>

@Html.DisplayFor(modelItem => item.Qty)

</td>

<td>

@Html.DisplayFor(modelItem => item.ExpiryDate)

</td>

<td>

@Html.ActionLink("Edit", "Edit", new { id=item.ProductID }) |

@Html.ActionLink("Details", "Details", new { id=item.ProductID }) |

@Html.ActionLink("Delete", "Delete", new { id=item.ProductID })

</td>

</tr>

}

</table>

**Edit:**

@model AssessmentQuestion.Models.Product

@{

ViewBag.Title = "Edit";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Edit</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>Product</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

@Html.HiddenFor(model => model.ProductID)

<div class="form-group">

@Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.Campany, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Campany, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Campany, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.Price, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Price, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Price, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.Qty, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Qty, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Qty, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.ExpiryDate, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.ExpiryDate, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.ExpiryDate, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<input type="submit" value="Save" class="btn btn-default" />

</div>

</div>

</div>

}

<div>

@Html.ActionLink("Back to List", "Index")

</div>

**Details:**

@model AssessmentQuestion.Models.Product

@{

ViewBag.Title = "Details";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Details</h2>

<div>

<h4>Product</h4>

<hr />

<dl class="dl-horizontal">

<dt>

@Html.DisplayNameFor(model => model.Name)

</dt>

<dd>

@Html.DisplayFor(model => model.Name)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Campany)

</dt>

<dd>

@Html.DisplayFor(model => model.Campany)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Price)

</dt>

<dd>

@Html.DisplayFor(model => model.Price)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Qty)

</dt>

<dd>

@Html.DisplayFor(model => model.Qty)

</dd>

<dt>

@Html.DisplayNameFor(model => model.ExpiryDate)

</dt>

<dd>

@Html.DisplayFor(model => model.ExpiryDate)

</dd>

</dl>

</div>

<p>

@Html.ActionLink("Edit", "Edit", new { id = Model.ProductID }) |

@Html.ActionLink("Back to List", "Index")

</p>

**Delete:**

@model AssessmentQuestion.Models.Product

@{

ViewBag.Title = "Delete";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Delete</h2>

<h3>Are you sure you want to delete this?</h3>

<div>

<h4>Product</h4>

<hr />

<dl class="dl-horizontal">

<dt>

@Html.DisplayNameFor(model => model.Name)

</dt>

<dd>

@Html.DisplayFor(model => model.Name)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Campany)

</dt>

<dd>

@Html.DisplayFor(model => model.Campany)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Price)

</dt>

<dd>

@Html.DisplayFor(model => model.Price)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Qty)

</dt>

<dd>

@Html.DisplayFor(model => model.Qty)

</dd>

<dt>

@Html.DisplayNameFor(model => model.ExpiryDate)

</dt>

<dd>

@Html.DisplayFor(model => model.ExpiryDate)

</dd>

</dl>

@using (Html.BeginForm()) {

@Html.AntiForgeryToken()

<div class="form-actions no-color">

<input type="submit" value="Delete" class="btn btn-default" /> |

@Html.ActionLink("Back to List", "Index")

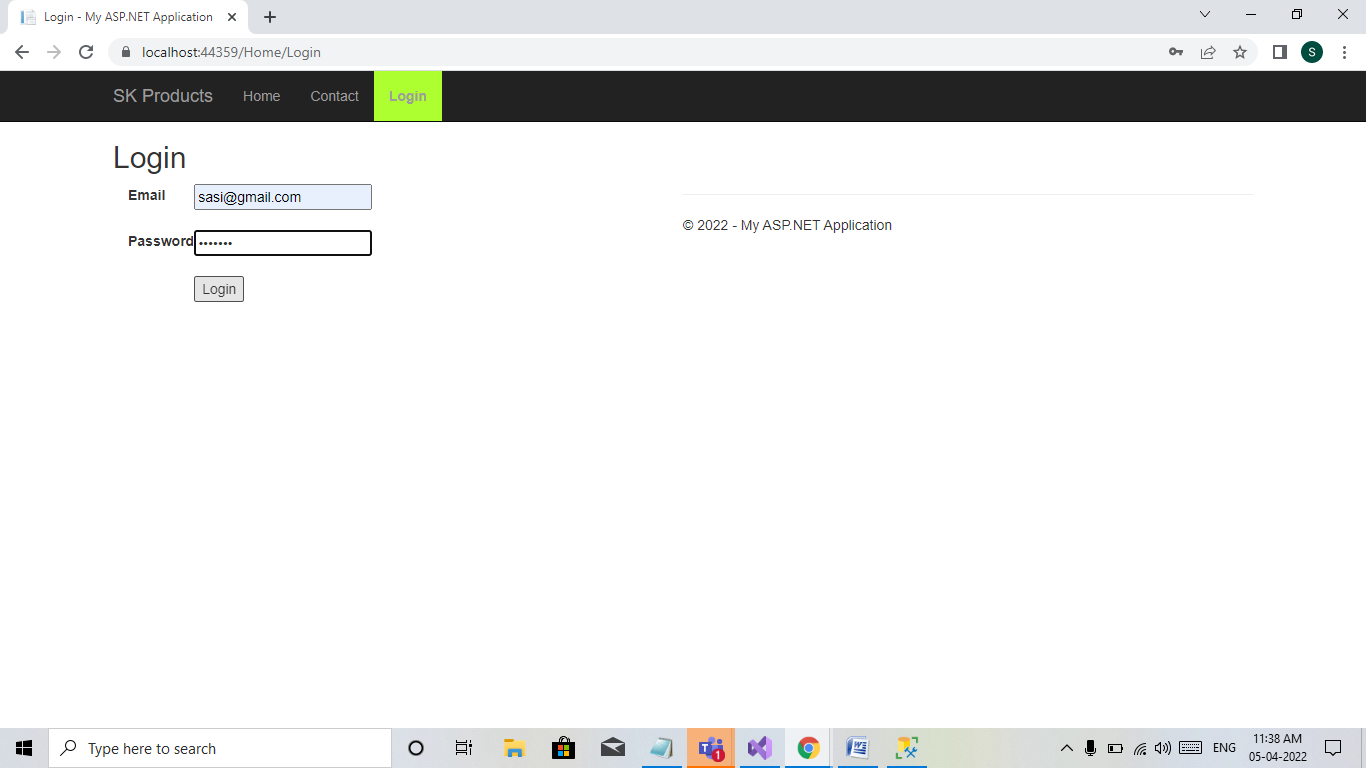
</div>

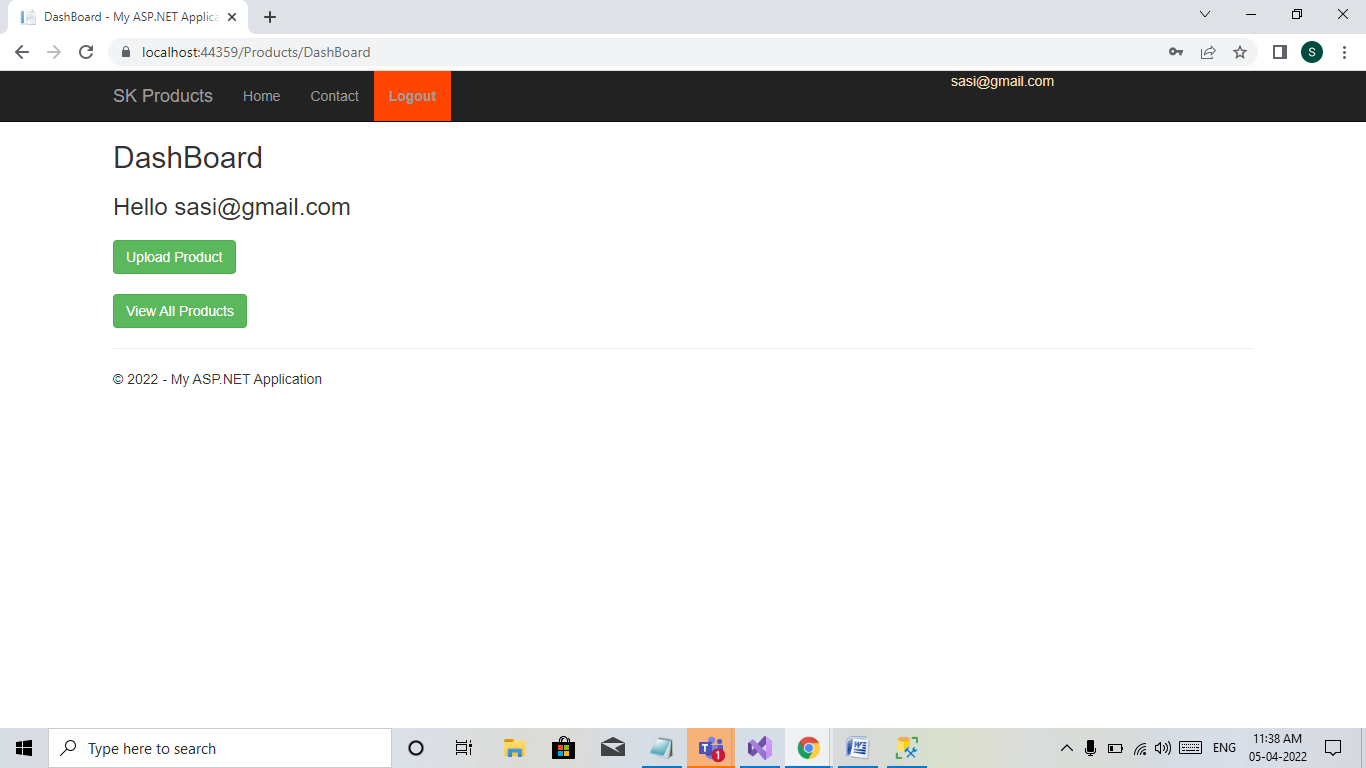
}

</div>

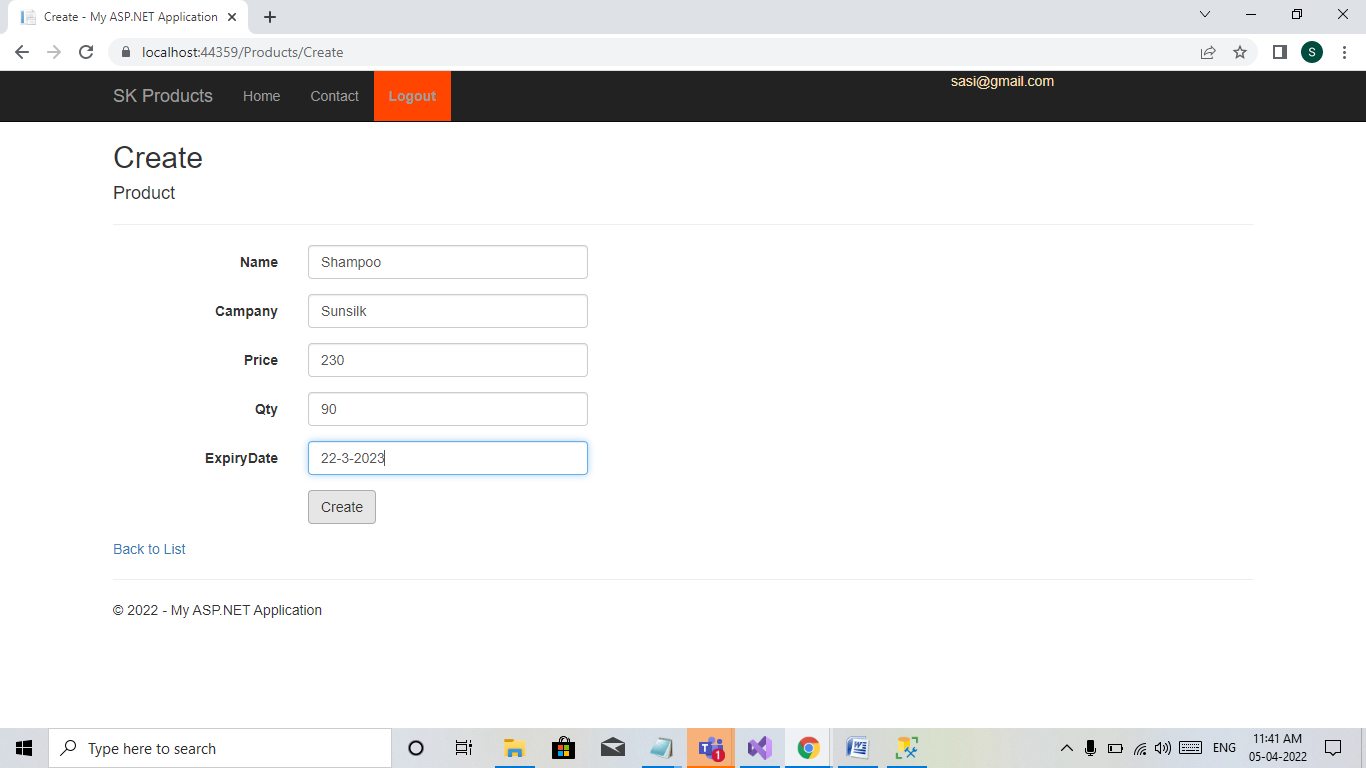
Output:

Login:

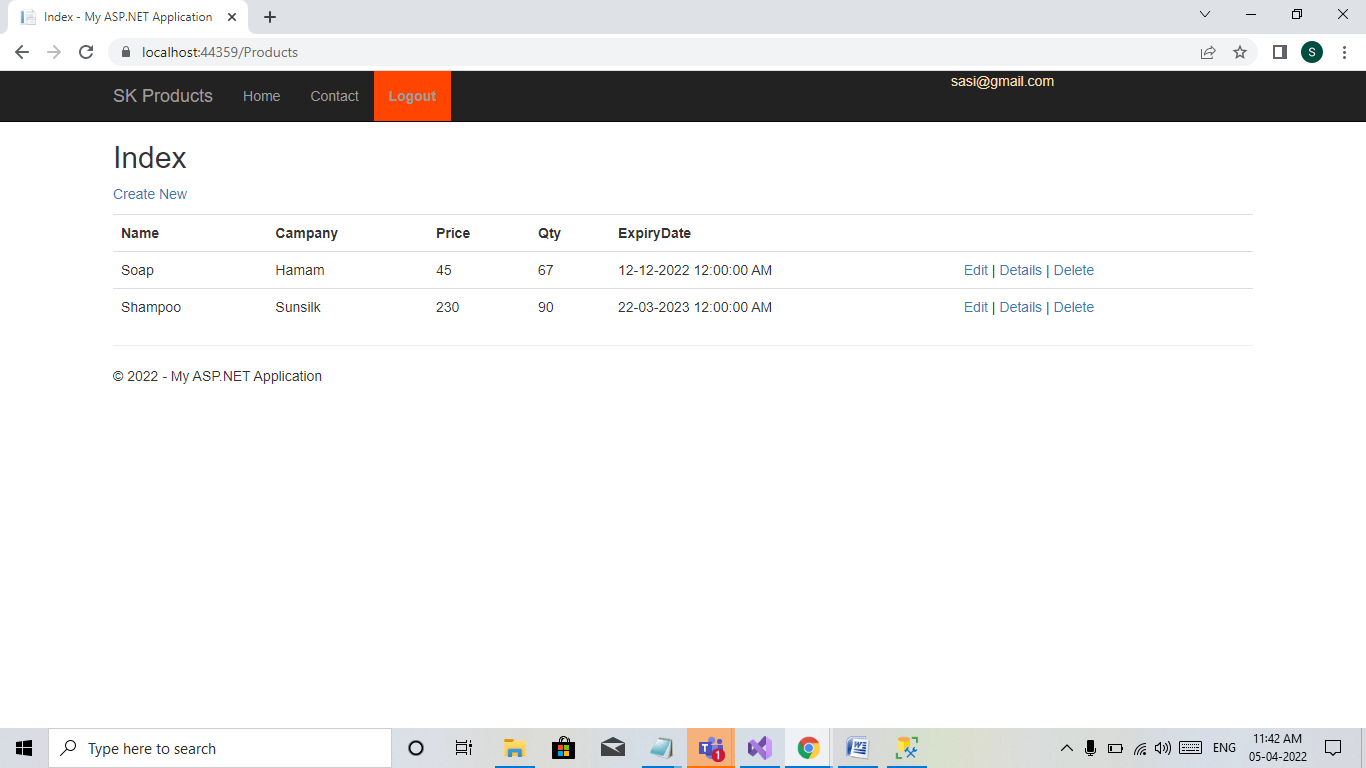


After Login DashBoard

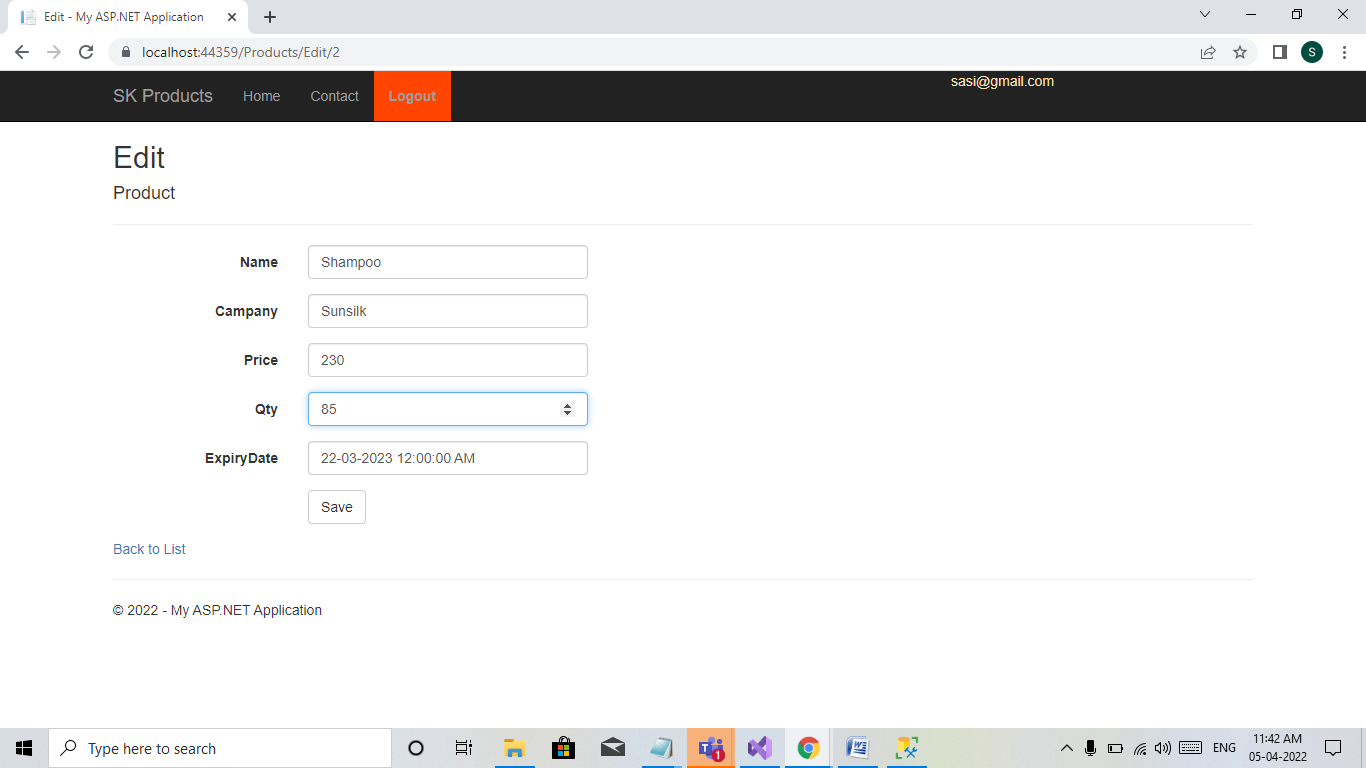
Email ID Maintained by session:. New Product Upload page



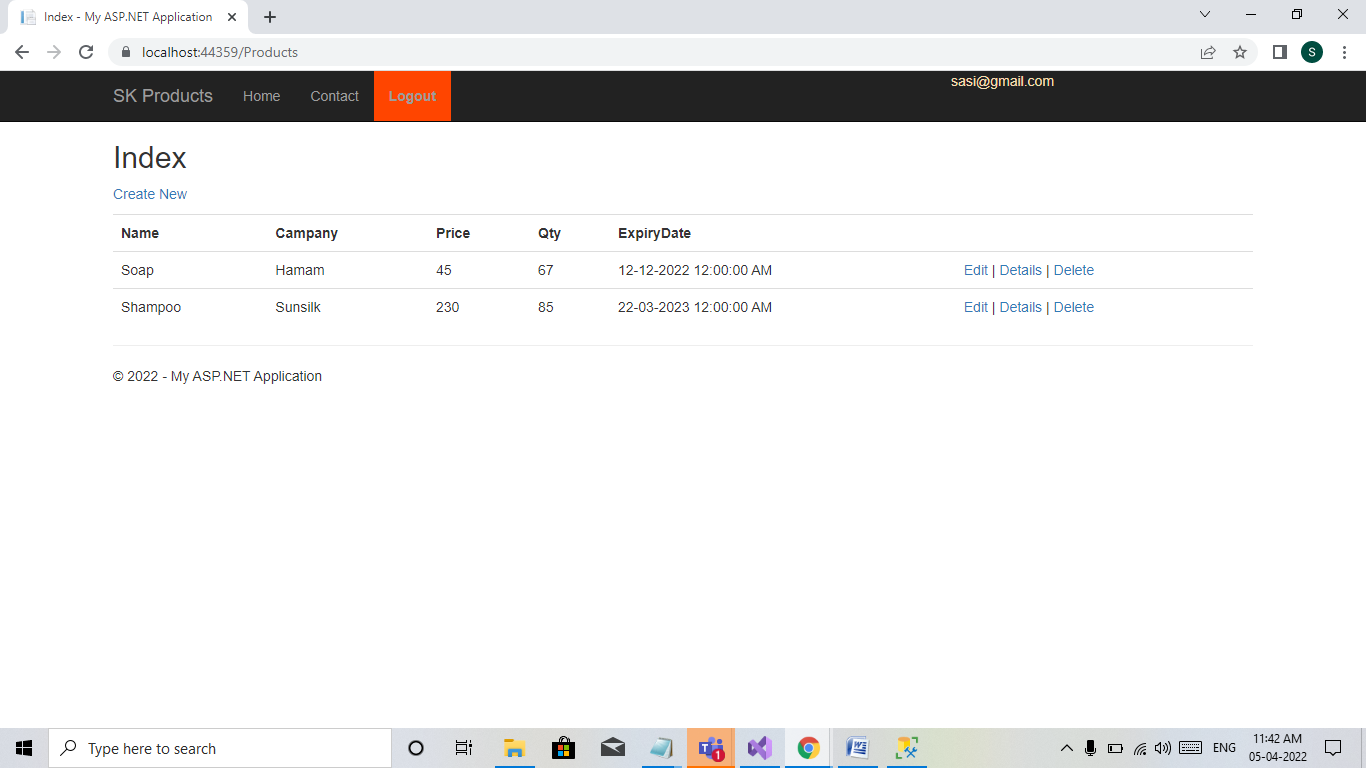
Index Page



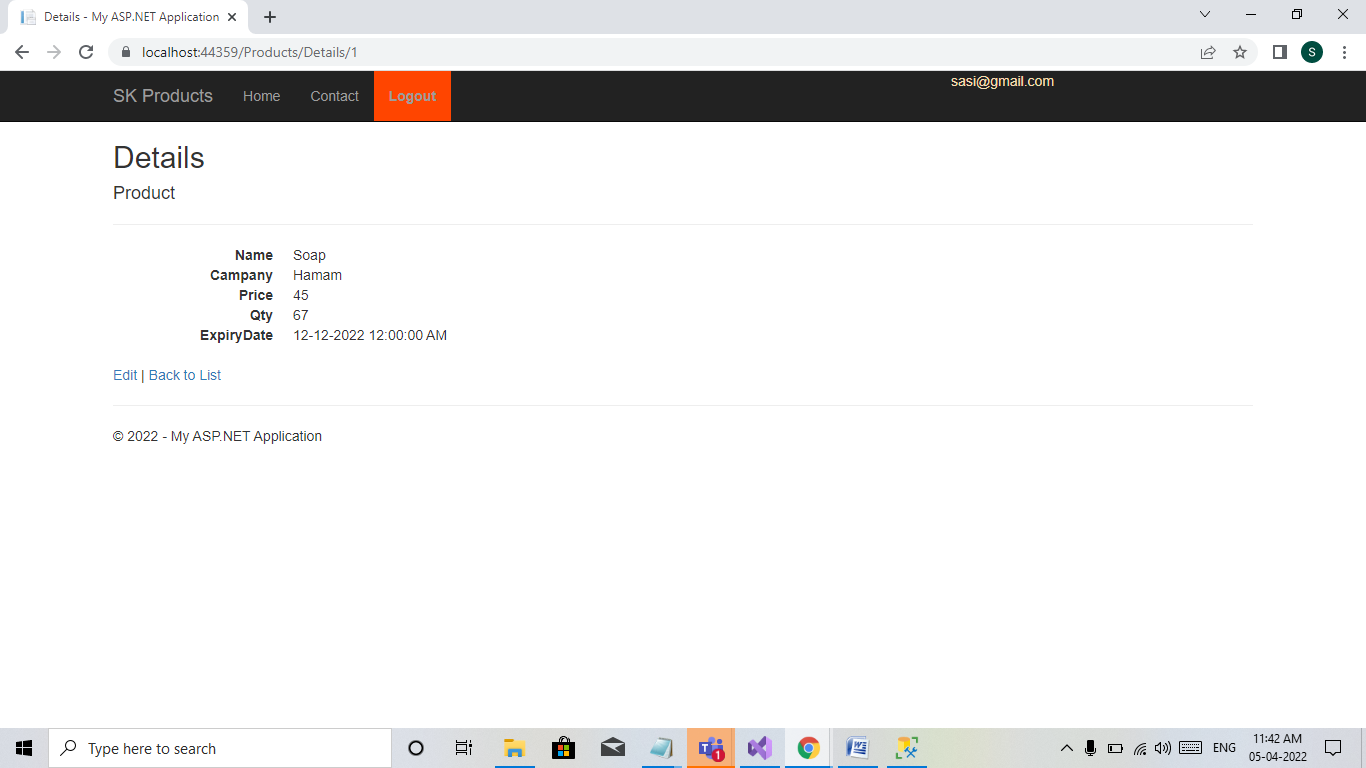
Edit Page

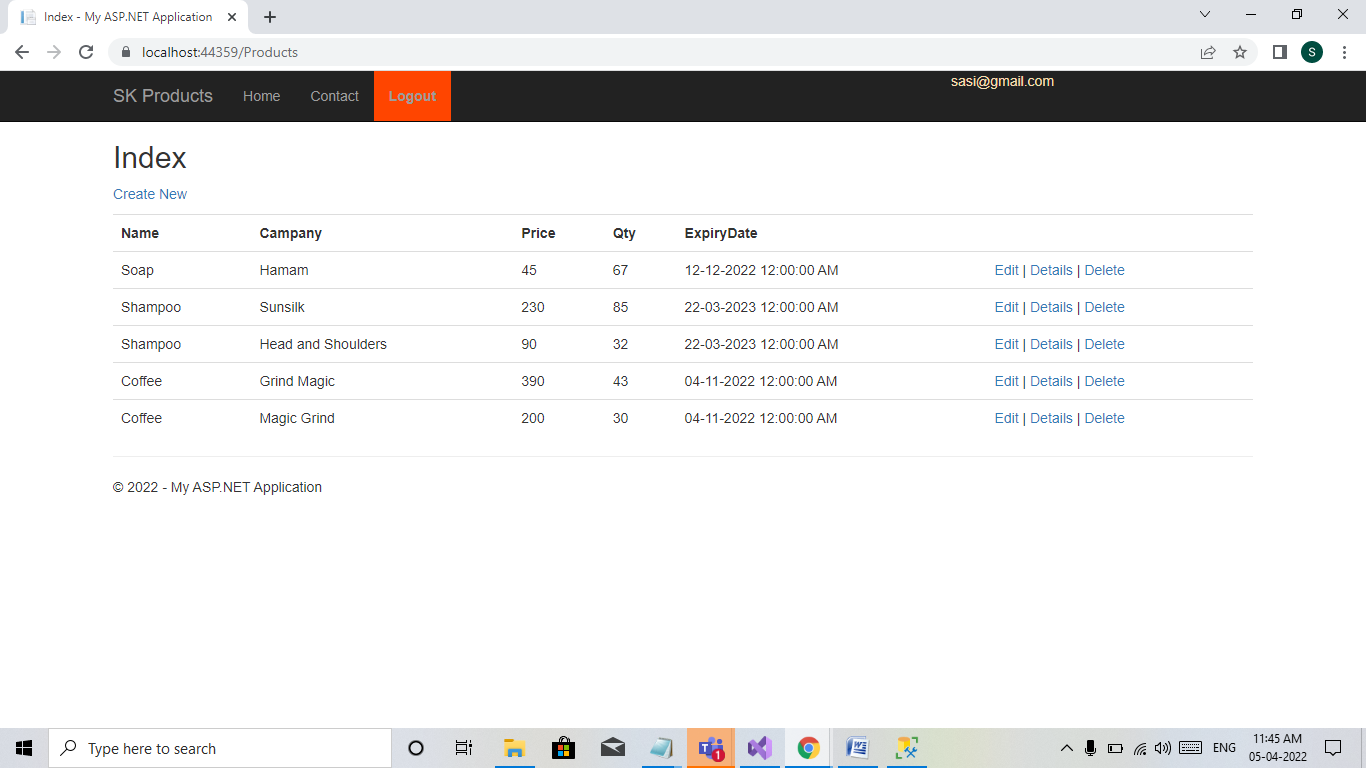


After Edit

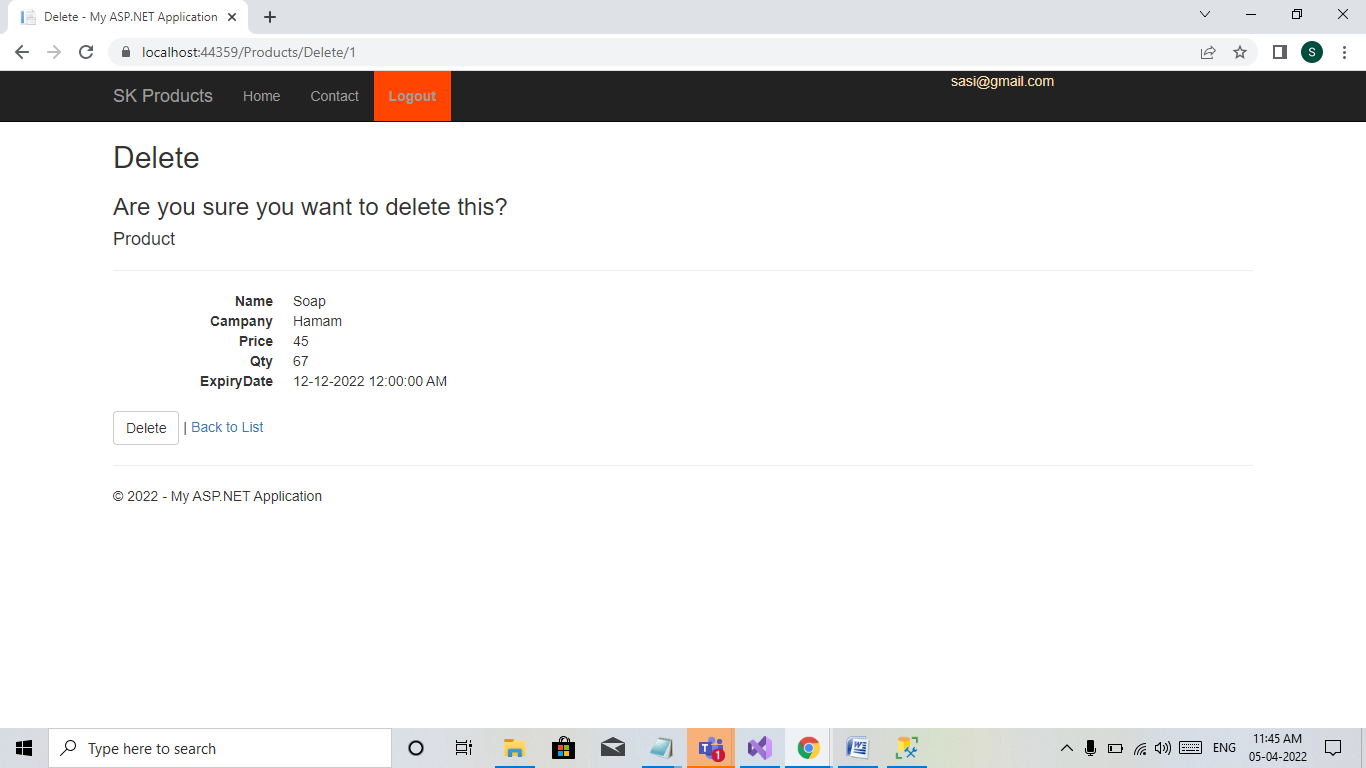


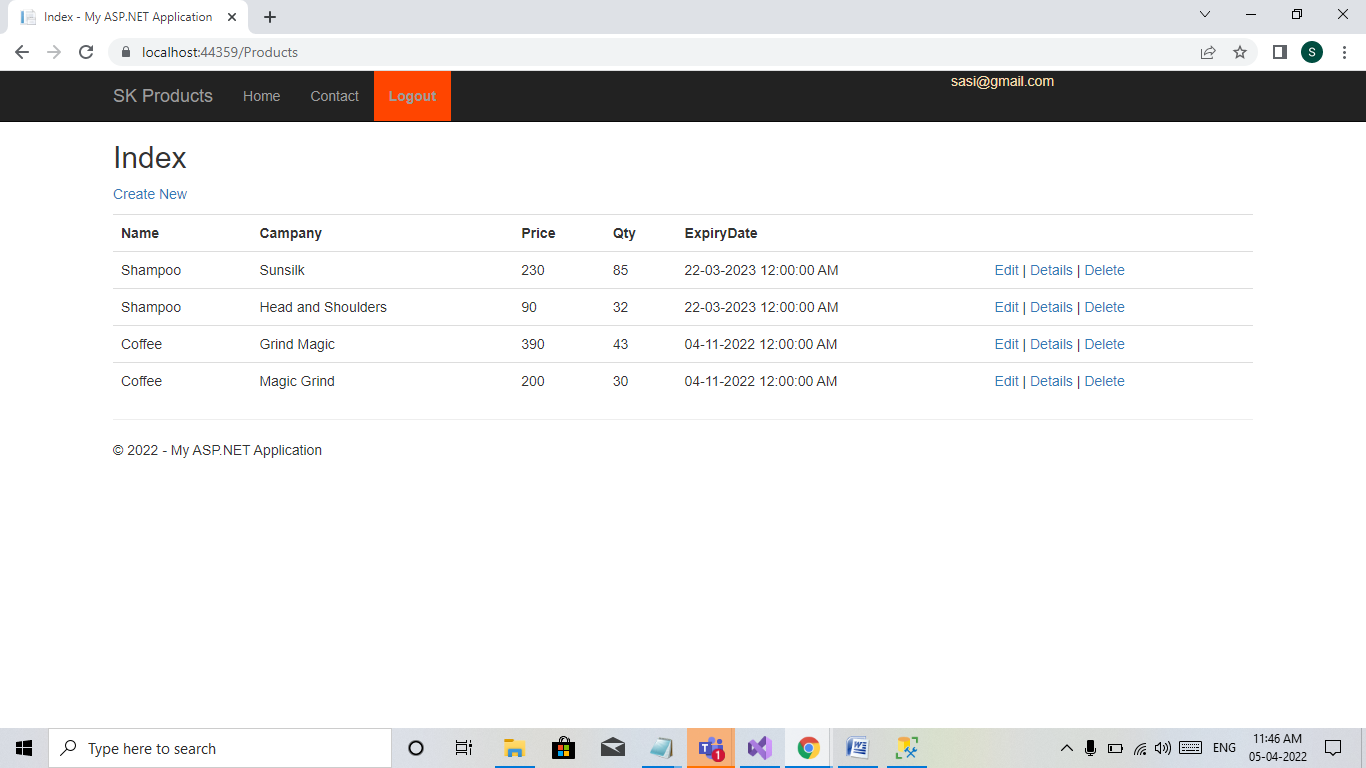
Details

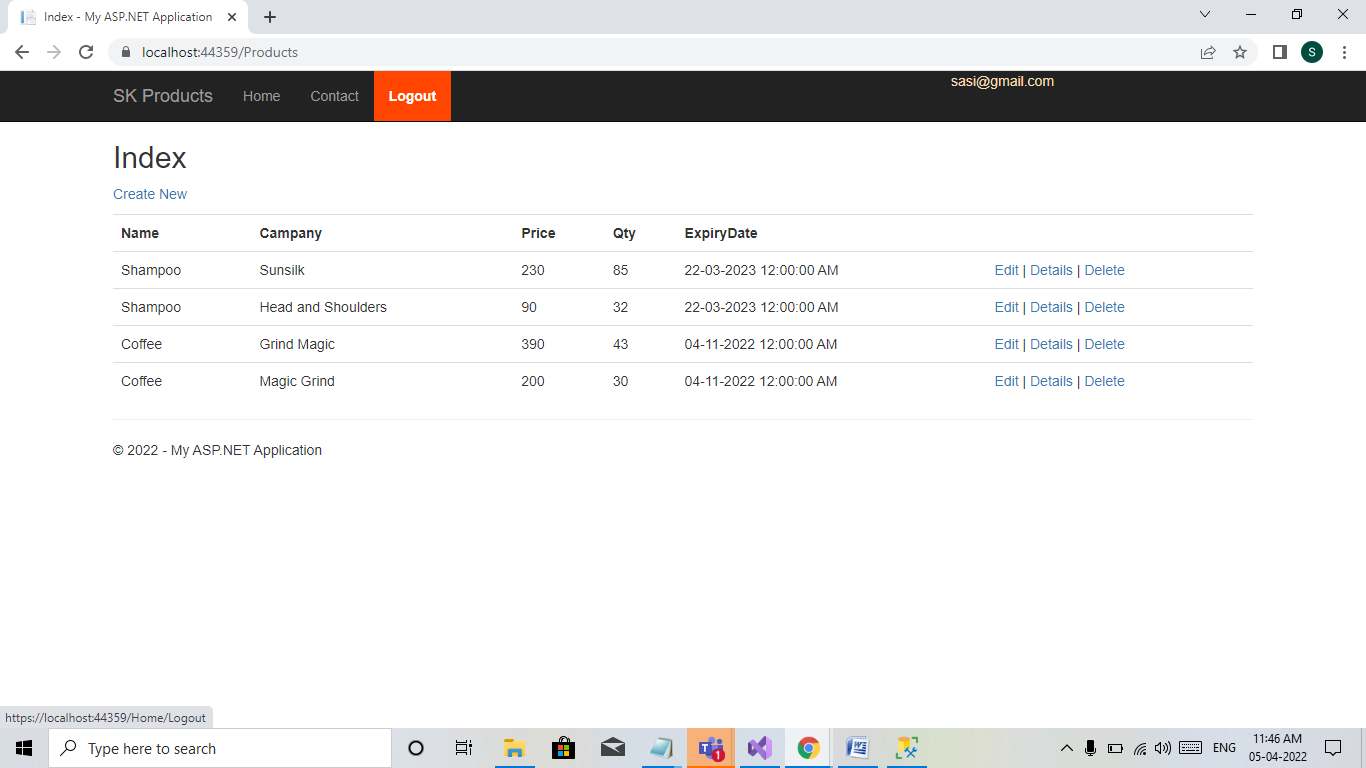




Delete Conformation page:







After Clicking Logout button session cleared and redirect to login page

