**MILESTONE ASSESSMENT 3**

**Design a kinder garten school management system where you can add student and staff, and you can assign students to staff. Admin should be able to select staff name from the dropdown and on selecting any staff name, Admin should be able to see the list of students assigned to the staff. Admin should be able to update staff information. Admin should be able to delete any student/Staff**

**CREATE TABLE**

CREATE TABLE [dbo].[Student]

(

[StudentId] INT NOT NULL PRIMARY KEY identity(1,1),

[StudentName] NVARCHAR(50) NULL,

[DateOfBirth] DATETIME NULL,

[ContactNo] BIGINT NULL,

[Address] NVARCHAR(200) NULL,

[StaffId] INT NULL foreign key references Staff(StaffId)

)

CREATE TABLE [dbo].[Staff]

(

[StaffId] INT NOT NULL PRIMARY KEY identity(1,1),

[StaffName] NVARCHAR(50) NULL,

[ContactNo] BIGINT NULL,

[EmailId] NVARCHAR(50) NULL,

[ClassType] NVARCHAR(50) NULL

)

CREATE TABLE [dbo].[standard]

(

[StandardId] INT NOT NULL PRIMARY KEY identity(1,1),

[StandardName] NVARCHAR(50) NULL,

)

CREATE TABLE [dbo].[credentials]

(

[id] INT NOT NULL PRIMARY KEY identity(1,1),

[username] NVARCHAR(50) NULL,

[password] NVARCHAR(50) NULL,

)

**CREATE PROCEDURE**

create procedure AddStaff(

@name nvarchar(50),

@contact int,

@email nvarchar(50),

@ctype nvarchar(50)

)

as

insert into Staff

(StaffName,ContactNO,EmailId,ClassType)

values(@name,@contact,@email,@ctype)

go

create procedure UpdateStaff(

@id int,

@contact int,

@email nvarchar(50)

)

as

update Staff

set ContactNO=@contact,

EmailId=@email

where StaffId=@id

go

create procedure AddStudent(

@name nvarchar(50),

@dob datetime,

@contact bigint,

@address nvarchar(200),

@stid int

)

as

insert into Student (StudentName,DateOfBirth,ContactNo,Address,StaffId)

values(@name,@dob,@contact,@address,@stid)

go

create procedure DeleteStudent(

@id int

)

as

delete from Student where StudentId=@id

go

**CREATE CONTROLLERS FOR ALL TABLES**

**CREDENTIALSCONTROLLER.CS CODE**

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 sms.Models;

namespace sms.Controllers

{

public class credentialsController : Controller

{

private school\_mgt\_sysEntities db = new school\_mgt\_sysEntities();

// GET: credentials

public ActionResult Index()

{

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

}

// GET: credentials/Details/5

public ActionResult Details(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

credential credential = db.credentials.Find(id);

if (credential == null)

{

return HttpNotFound();

}

return View(credential);

}

// GET: credentials/Create

public ActionResult Create()

{

return View();

}

// POST: credentials/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 = "id,username,password")] credential credential)

{

if (ModelState.IsValid)

{

db.credentials.Add(credential);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(credential);

}

// GET: credentials/Edit/5

public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

credential credential = db.credentials.Find(id);

if (credential == null)

{

return HttpNotFound();

}

return View(credential);

}

// POST: credentials/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 = "id,username,password")] credential credential)

{

if (ModelState.IsValid)

{

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

db.SaveChanges();

return RedirectToAction("Index");

}

return View(credential);

}

// GET: credentials/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

credential credential = db.credentials.Find(id);

if (credential == null)

{

return HttpNotFound();

}

return View(credential);

}

// POST: credentials/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

credential credential = db.credentials.Find(id);

db.credentials.Remove(credential);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}

}

**HOME CONTROLLER.CS CODE**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace sms.Controllers

{

public class HomeController : Controller

{

public ActionResult Index()

{

return View();

}

public ActionResult About()

{

ViewBag.Message = "Your application description page.";

return View();

}

public ActionResult Contact()

{

ViewBag.Message = "Your contact page.";

return View();

}

}

}

**STAFFSCONTROLLER.CS CODE**

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 sms.Models;

namespace sms.Controllers

{

public class StaffsController : Controller

{

private school\_mgt\_sysEntities db = new school\_mgt\_sysEntities();

// GET: Staffs

public ActionResult Index()

{

var staffs = db.Staffs.Include(s => s.standard);

return View(staffs.ToList());

}

// GET: Staffs/Details/5

public ActionResult Details(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Staff staff = db.Staffs.Find(id);

if (staff == null)

{

return HttpNotFound();

}

return View(staff);

}

// GET: Staffs/Create

public ActionResult Create()

{

ViewBag.StandardType = new SelectList(db.standards, "standard\_id", "standard\_name");

return View();

}

// POST: Staffs/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 = "StaffId,StaffName,ContactNo,EmailId,StandardType")] Staff staff)

{

if (ModelState.IsValid)

{

db.Staffs.Add(staff);

db.SaveChanges();

return RedirectToAction("Index");

}

ViewBag.StandardType = new SelectList(db.standards, "standard\_id", "standard\_name", staff.StandardType);

return View(staff);

}

// GET: Staffs/Edit/5

public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Staff staff = db.Staffs.Find(id);

if (staff == null)

{

return HttpNotFound();

}

ViewBag.StandardType = new SelectList(db.standards, "standard\_id", "standard\_name", staff.StandardType);

return View(staff);

}

// POST: Staffs/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 = "StaffId,StaffName,ContactNo,EmailId,StandardType")] Staff staff)

{

if (ModelState.IsValid)

{

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

db.SaveChanges();

return RedirectToAction("Index");

}

ViewBag.StandardType = new SelectList(db.standards, "standard\_id", "standard\_name", staff.StandardType);

return View(staff);

}

// GET: Staffs/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Staff staff = db.Staffs.Find(id);

if (staff == null)

{

return HttpNotFound();

}

return View(staff);

}

// POST: Staffs/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Staff staff = db.Staffs.Find(id);

db.Staffs.Remove(staff);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}

}

**STANDARDSCONTROLLER.CS CODE**

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 sms.Models;

namespace sms.Controllers

{

public class standardsController : Controller

{

private school\_mgt\_sysEntities db = new school\_mgt\_sysEntities();

// GET: standards

public ActionResult Index()

{

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

}

// GET: standards/Details/5

public ActionResult Details(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

standard standard = db.standards.Find(id);

if (standard == null)

{

return HttpNotFound();

}

return View(standard);

}

// GET: standards/Create

public ActionResult Create()

{

return View();

}

// POST: standards/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 = "standard\_id,standard\_name")] standard standard)

{

if (ModelState.IsValid)

{

db.standards.Add(standard);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(standard);

}

// GET: standards/Edit/5

public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

standard standard = db.standards.Find(id);

if (standard == null)

{

return HttpNotFound();

}

return View(standard);

}

// POST: standards/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 = "standard\_id,standard\_name")] standard standard)

{

if (ModelState.IsValid)

{

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

db.SaveChanges();

return RedirectToAction("Index");

}

return View(standard);

}

// GET: standards/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

standard standard = db.standards.Find(id);

if (standard == null)

{

return HttpNotFound();

}

return View(standard);

}

// POST: standards/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

standard standard = db.standards.Find(id);

db.standards.Remove(standard);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}

}

**STUDENTSCONTROLLER.CS CODE**

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 sms.Models;

namespace sms.Controllers

{

public class credentialsController : Controller

{

private school\_mgt\_sysEntities db = new school\_mgt\_sysEntities();

// GET: credentials

public ActionResult Index()

{

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

}

// GET: credentials/Details/5

public ActionResult Details(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

credential credential = db.credentials.Find(id);

if (credential == null)

{

return HttpNotFound();

}

return View(credential);

}

// GET: credentials/Create

public ActionResult Create()

{

return View();

}

// POST: credentials/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 = "id,username,password")] credential credential)

{

if (ModelState.IsValid)

{

db.credentials.Add(credential);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(credential);

}

// GET: credentials/Edit/5

public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

credential credential = db.credentials.Find(id);

if (credential == null)

{

return HttpNotFound();

}

return View(credential);

}

// POST: credentials/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 = "id,username,password")] credential credential)

{

if (ModelState.IsValid)

{

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

db.SaveChanges();

return RedirectToAction("Index");

}

return View(credential);

}

// GET: credentials/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

credential credential = db.credentials.Find(id);

if (credential == null)

{

return HttpNotFound();

}

return View(credential);

}

// POST: credentials/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

credential credential = db.credentials.Find(id);

db.credentials.Remove(credential);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}

}

In shared \_Layout.cshtml write this code:

<!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("Application name", "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("Staffs", "Index", "Staffs")</li>

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

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

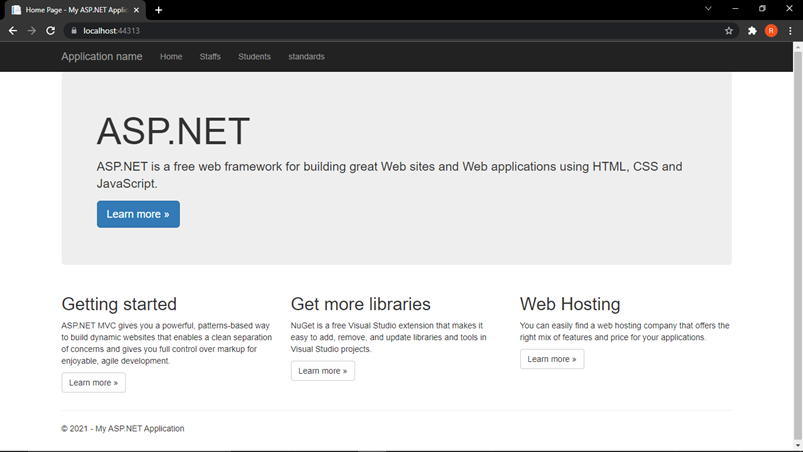
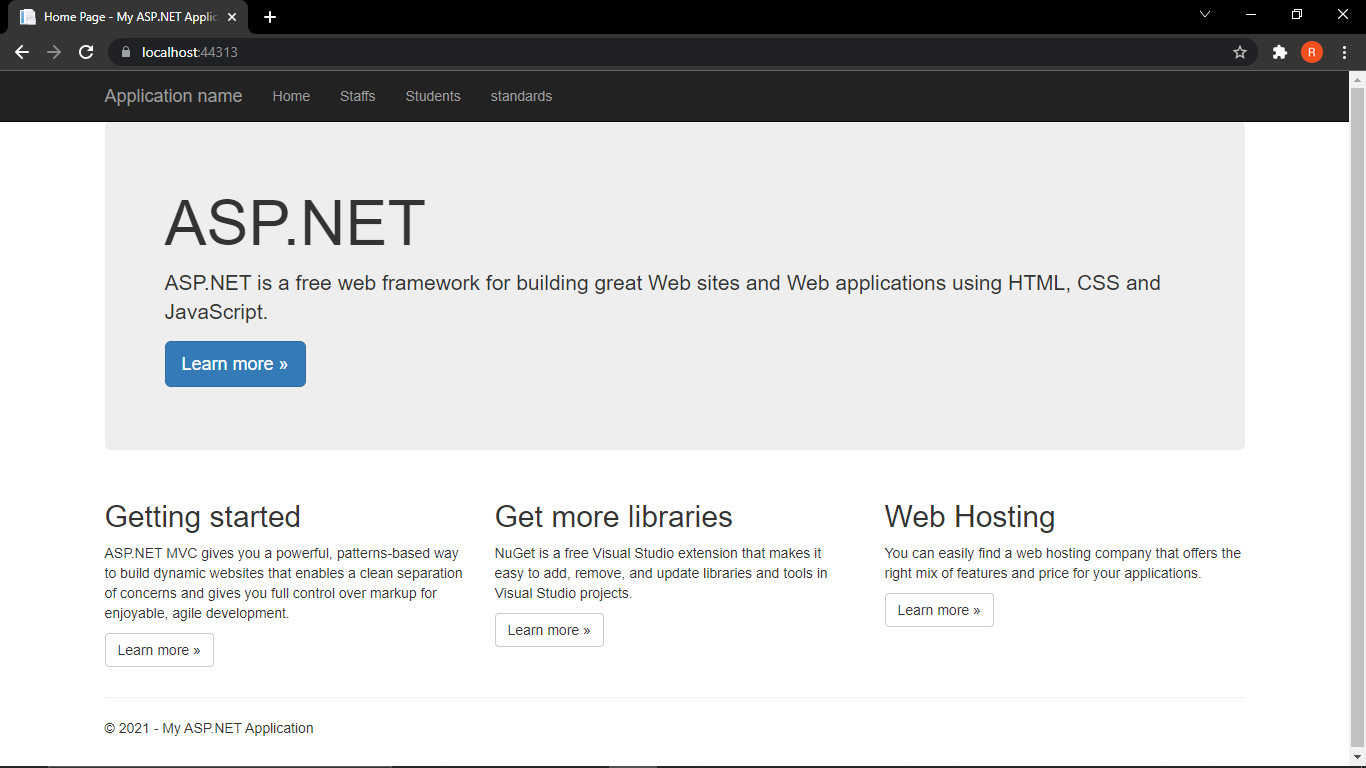
</body>

</html>

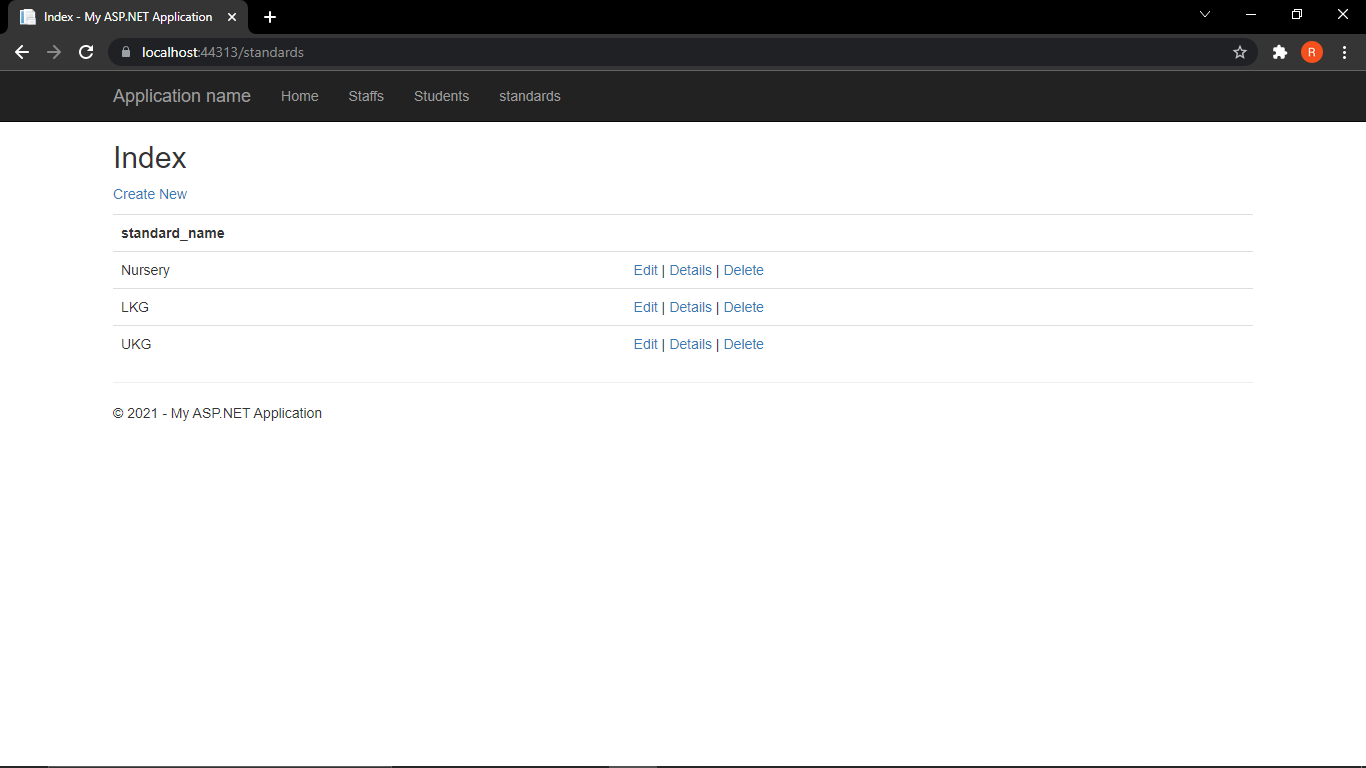
**THEN RUN THE APPLICATION**

**OUTPUT**

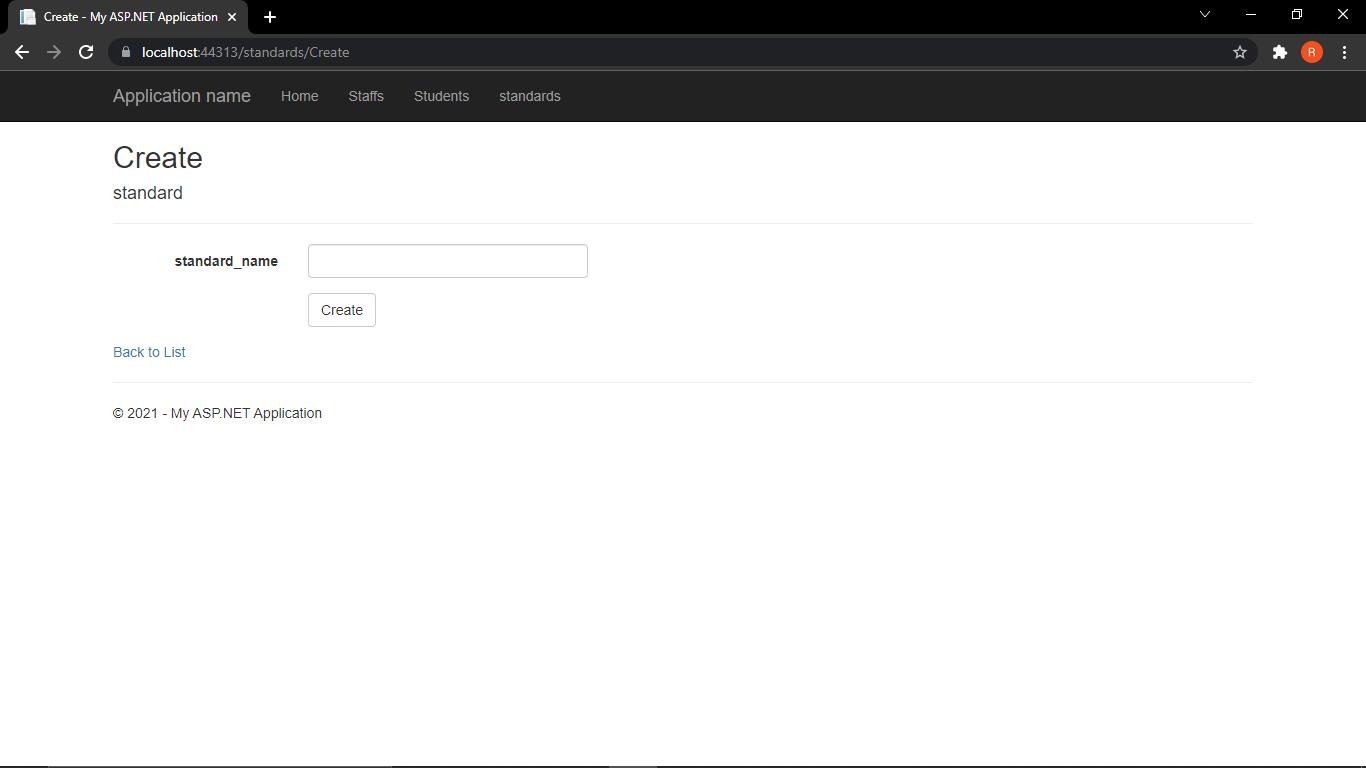
**HOMEPAGE:**



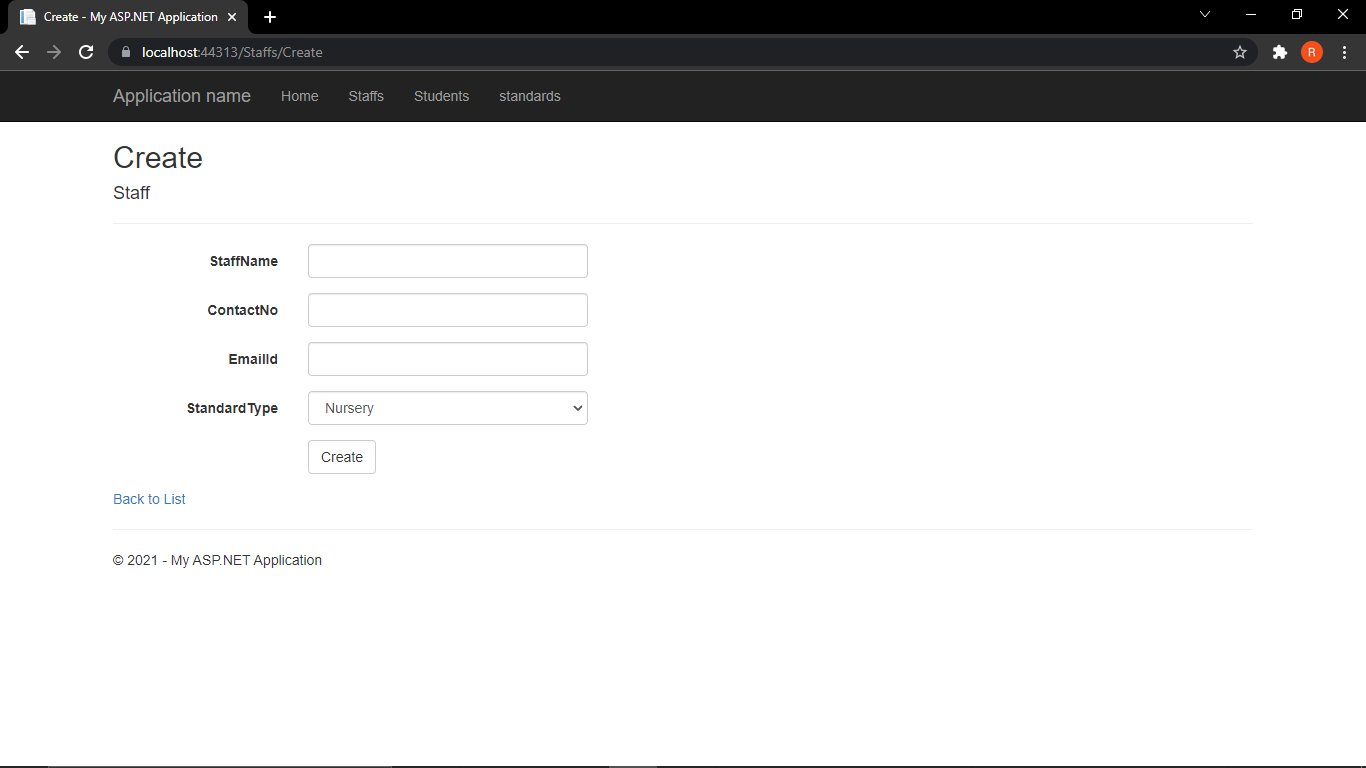
**STANDARD(CLASS PAGE)**



**STANDARD PAGE**



**STAFF PAGE**



**STUDENT PAGE**

