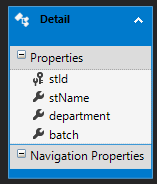
**LAB # 04**

**Task # 01: Create CRUD application using ado.net.**

**Solution:**



**Controller:**

namespace WebApplication5.Controllers

{

public class DetailsController : Controller

{

private StudentEntities db = new StudentEntities();

// GET: Details

public ActionResult Index()

{

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

}

// GET: Details/Details/5

public ActionResult Details(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Detail detail = db.Details.Find(id);

if (detail == null)

{

return HttpNotFound();

}

return View(detail);

}

// GET: Details/Create

public ActionResult Create()

{

return View();

}

// POST: Details/Create

// To protect from overposting attacks, please 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 = "stId,stName,department,batch")] Detail detail)

{

if (ModelState.IsValid)

{

db.Details.Add(detail);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(detail);

}

// GET: Details/Edit/5

public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Detail detail = db.Details.Find(id);

if (detail == null)

{

return HttpNotFound();

}

return View(detail);

}

// POST: Details/Edit/5

// To protect from overposting attacks, please 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 = "stId,stName,department,batch")] Detail detail)

{

if (ModelState.IsValid)

{

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

db.SaveChanges();

return RedirectToAction("Index");

}

return View(detail);

}

// GET: Details/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Detail detail = db.Details.Find(id);

if (detail == null)

{

return HttpNotFound();

}

return View(detail);

}

// POST: Details/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Detail detail = db.Details.Find(id);

db.Details.Remove(detail);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}

}

**Index.cshtml:**

@model IEnumerable<WebApplication5.Models.Detail>

@{

ViewBag.Title = "Index";

}

<h2>Index</h2>

<p>

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

</p>

<table class="table">

<tr>

<th>

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

</th>

<th>

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

</th>

<th>

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

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

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

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

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

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

</td>

</tr>

}

</table>

**Create.cshtml:**

@model WebApplication5.Models.Detail

@{

ViewBag.Title = "Create";

}

<h2>Create</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>Detail</h4>

<hr />

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

<div class="form-group">

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

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

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

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

</div>

</div>

<div class="form-group">

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

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

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

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

</div>

</div>

<div class="form-group">

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

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

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

@Html.ValidationMessageFor(model => model.batch, "", 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>

@section Scripts {

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

}

**Delete.cshtml:**

@model WebApplication5.Models.Detail

@{

ViewBag.Title = "Delete";

}

<h2>Delete</h2>

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

<div>

<h4>Detail</h4>

<hr />

<dl class="dl-horizontal">

<dt>

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

</dt>

<dd>

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

</dd>

<dt>

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

</dt>

<dd>

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

</dd>

<dt>

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

</dt>

<dd>

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

</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>

**Detail.cshtml:**

@model WebApplication5.Models.Detail

@{

ViewBag.Title = "Details";

}

<h2>Details</h2>

<div>

<h4>Detail</h4>

<hr />

<dl class="dl-horizontal">

<dt>

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

</dt>

<dd>

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

</dd>

<dt>

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

</dt>

<dd>

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

</dd>

<dt>

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

</dt>

<dd>

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

</dd>

</dl>

</div>

<p>

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

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

</p>

**Edit.cshtml:**

@model WebApplication5.Models.Detail

@{

ViewBag.Title = "Edit";

}

<h2>Edit</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>Detail</h4>

<hr />

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

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

<div class="form-group">

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

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

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

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

</div>

</div>

<div class="form-group">

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

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

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

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

</div>

</div>

<div class="form-group">

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

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

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

@Html.ValidationMessageFor(model => model.batch, "", 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>

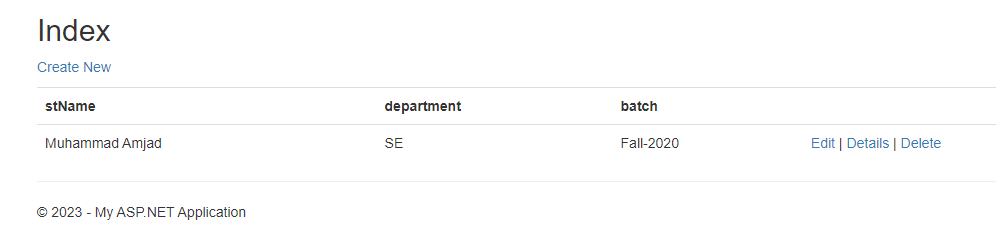
@section Scripts {

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

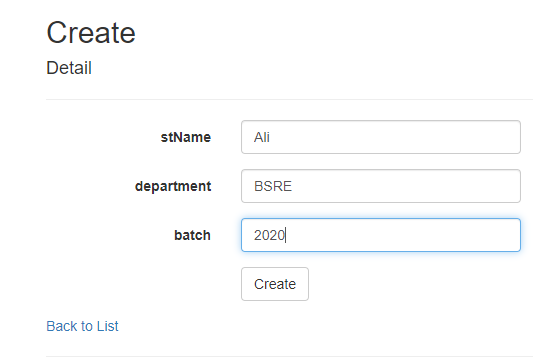
}

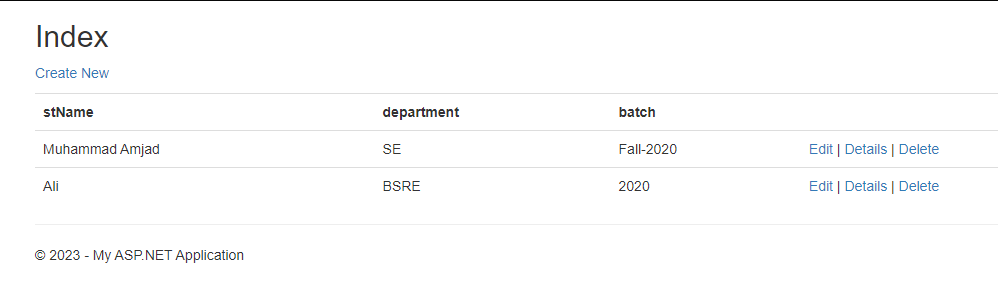
**Output:**

**Index:**

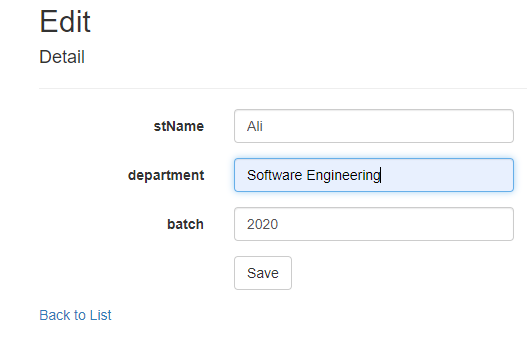


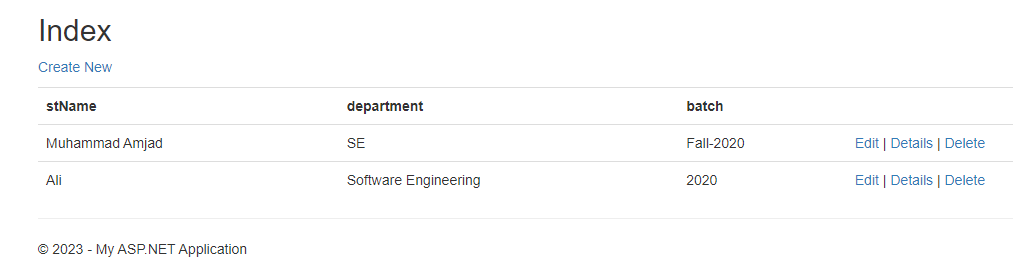
**Create:**



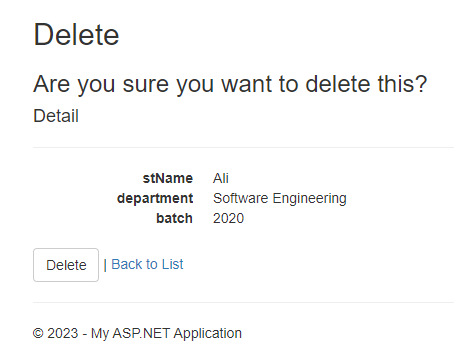


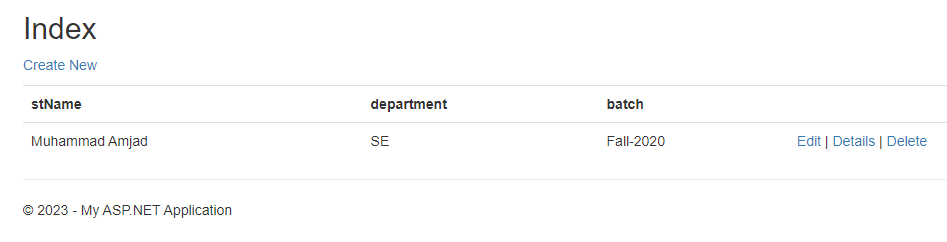
**Edit:**



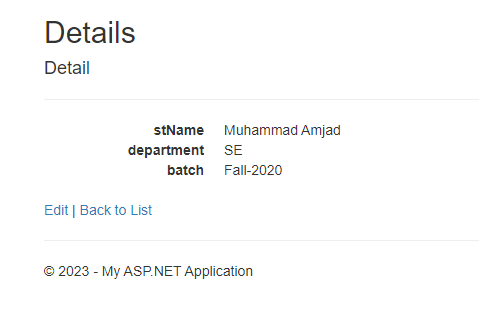


**Delete:**





**Detail:**



**Task # 02: Create Book Detail Store.**

**Solution:**

**Controller:**

public class BookController : Controller

{

public IActionResult Index()

{

return View();

}

[HttpPost]

public IActionResult Index(Book book)

{

if (ModelState.IsValid)

{

return View(book);

}

return View();

}

}

**Model:**

namespace WebApplication8.Models

{

public class Book

{

public string title { get; set; }

public string author { get; set; }

public string publisher { get; set; }

}

}

**Index.cshtml:**

@model WebApplication8.Models.Book

@{

ViewData["Title"] = "Index";

}

<h1>Index</h1>

<h4>Book</h4>

<hr />

<div class="row">

<div class="col-md-4">

<**form** **asp-action**="Index">

<**div** **asp-validation-summary**="ModelOnly" class="text-danger"></**div**>

<div class="form-group">

<**label** **asp-for**="title" class="control-label"></**label**>

<**input** **asp-for**="title" class="form-control" />

<**span** **asp-validation-for**="title" class="text-danger"></**span**>

</div><br />

<div class="form-group">

<**label** **asp-for**="author" class="control-label"></**label**>

<**input** **asp-for**="author" class="form-control" />

<**span** **asp-validation-for**="author" class="text-danger"></**span**>

</div><br />

<div class="form-group">

<**label** **asp-for**="publisher" class="control-label"></**label**>

<**input** **asp-for**="publisher" class="form-control" />

<**span** **asp-validation-for**="publisher" class="text-danger"></**span**>

</div><br />

<div class="form-group">

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

</div>

</**form**>

</div>

</div>

@if (Model != null)

{

<h4>Book Details:</h4>

<hr />

<div>

<label>Title: @Model.title </label><br />

<label>Author: @Model.author</label><br />

<label>Publisher: @Model.publisher</label><br />

</div>

}

@section Scripts {

@{await Html.RenderPartialAsync("\_ValidationScriptsPartial");}

}

**Output:** 