**Q-1. Create an MVC(.net framework or Core) application with C# Diamond information using entity framework. Convert EF to tuple collection and perform CRUD operation on it.**

**Ans:-**

**Diamondscontroller.cs**

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 PraticalAssignment;

namespace PraticalAssignment.Controllers

{

public class DiamondsController : Controller

{

private DiamondDBEntities db = new DiamondDBEntities();

// GET: Diamonds

public ActionResult Index()

{

List<Diamond> diamonds = db.Diamonds.ToList();

var tuples = Tuple.Create<List<Diamond>>(diamonds);

return View(tuples.Item1);

}

// GET: Diamonds/Details/5

public ActionResult Details(int? id)

{

Diamond diamonds = db.Diamonds.Find(id);

var tuples = Tuple.Create<Diamond>(diamonds);

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

if (tuples.Item1 == null)

{

return HttpNotFound();

}

return View(tuples.Item1);

}

// GET: Diamonds/Create

public ActionResult Create()

{

return View();

}

// POST: Diamonds/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 = "diamondId,diamondName,Weight,ColorGrade,ClarityGrade,CutGrade")] Diamond diamond)

{

if (ModelState.IsValid)

{

db.Diamonds.Add(diamond);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(diamond);

}

// GET: Diamonds/Edit/5

public ActionResult Edit(int? id)

{

Diamond diamonds = db.Diamonds.Find(id);

var tuples = Tuple.Create<Diamond>(diamonds);

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

if (tuples.Item1 == null)

{

return HttpNotFound();

}

return View(tuples.Item1);

}

// POST: Diamonds/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 = "diamondId,diamondName,Weight,ColorGrade,ClarityGrade,CutGrade")] Diamond diamond)

{

if (ModelState.IsValid)

{

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

db.SaveChanges();

return RedirectToAction("Index");

}

return View(diamond);

}

// GET: Diamonds/Delete/5

public ActionResult Delete(int? id)

{

Diamond diamonds = db.Diamonds.Find(id);

var tuples = Tuple.Create<Diamond>(diamonds);

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

if (tuples.Item1 == null)

{

return HttpNotFound();

}

return View(tuples.Item1);

}

// POST: Diamonds/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Diamond diamonds = db.Diamonds.Find(id);

var tuples = Tuple.Create<Diamond>(diamonds);

db.Diamonds.Remove(tuples.Item1);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}

}

**Diamonds.cs**

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated from a template.

//

// Manual changes to this file may cause unexpected behavior in your application.

// Manual changes to this file will be overwritten if the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

namespace PraticalAssignment

{

using System;

using System.Collections.Generic;

public partial class Diamond

{

public int diamondId { get; set; }

public string diamondName { get; set; }

public double Weight { get; set; }

public string ColorGrade { get; set; }

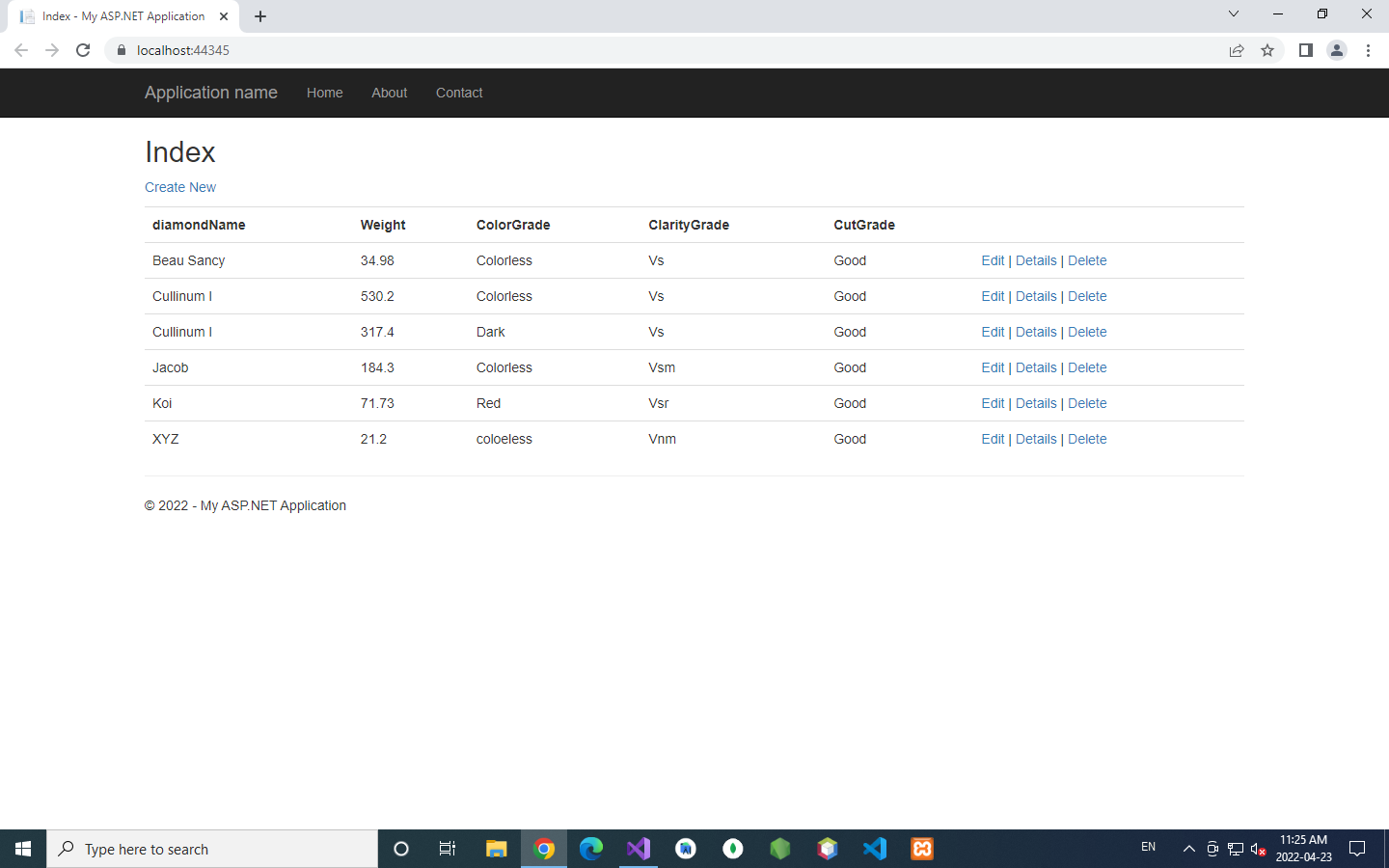
public string ClarityGrade { get; set; }

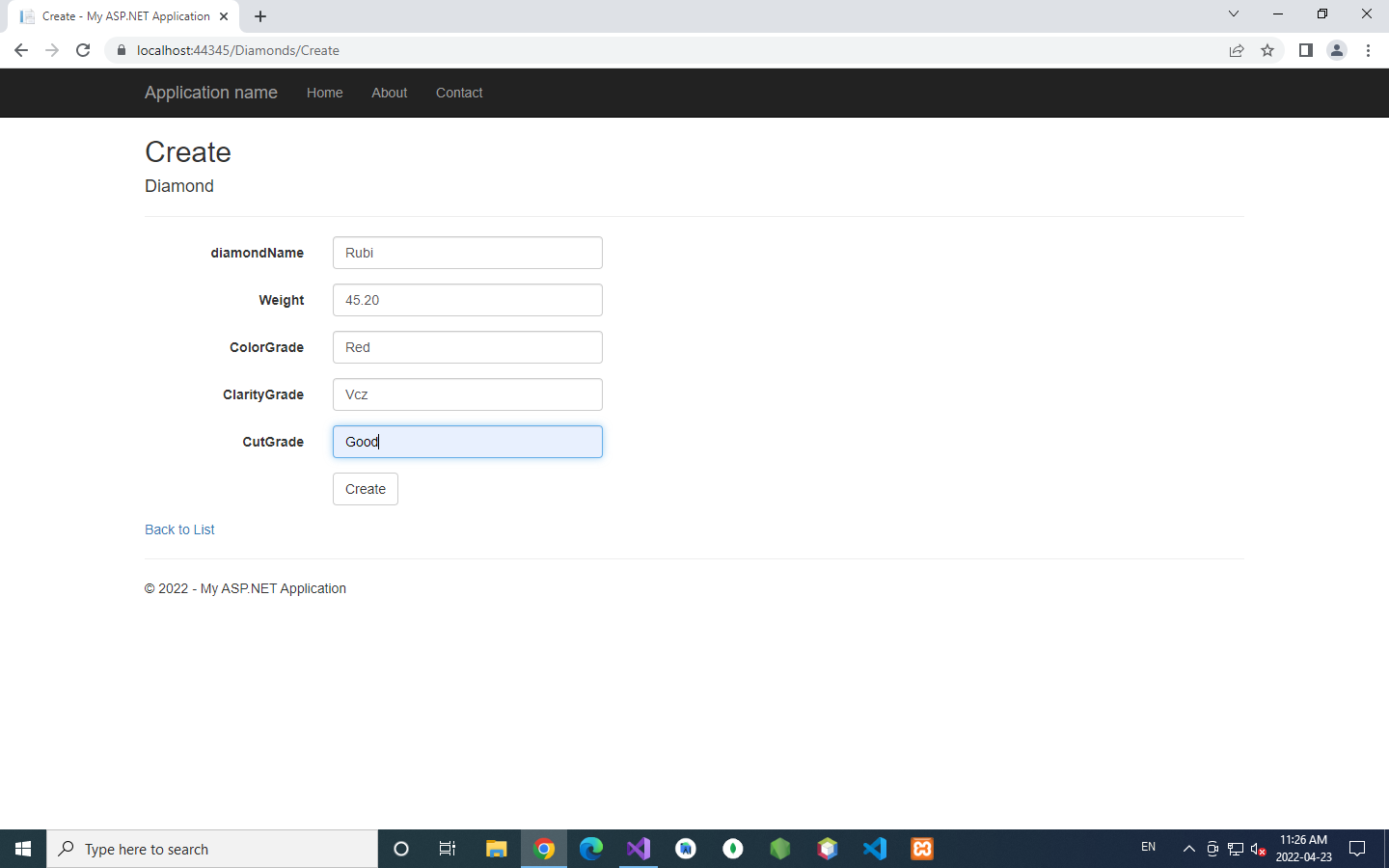
public string CutGrade { get; set; }

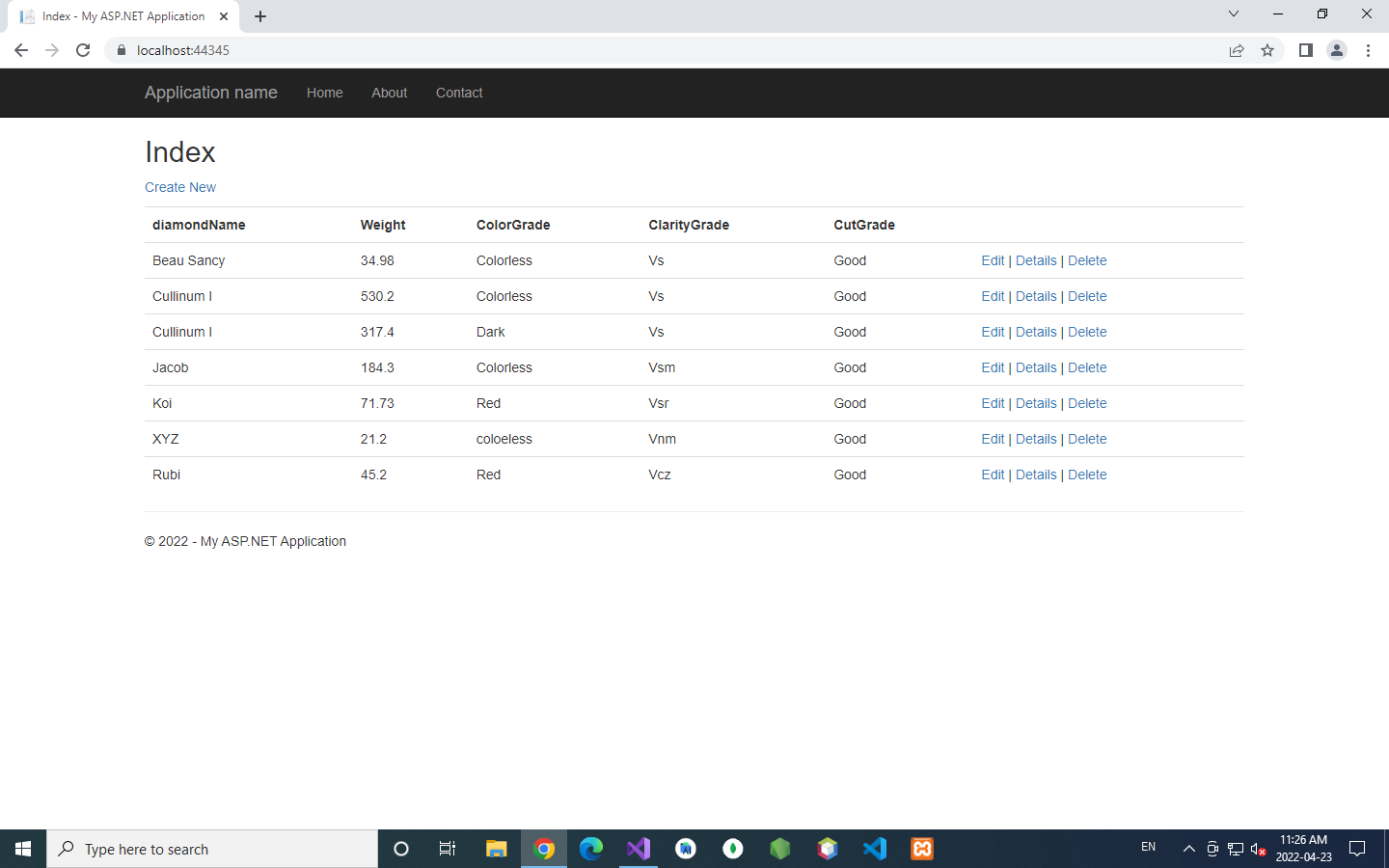
}

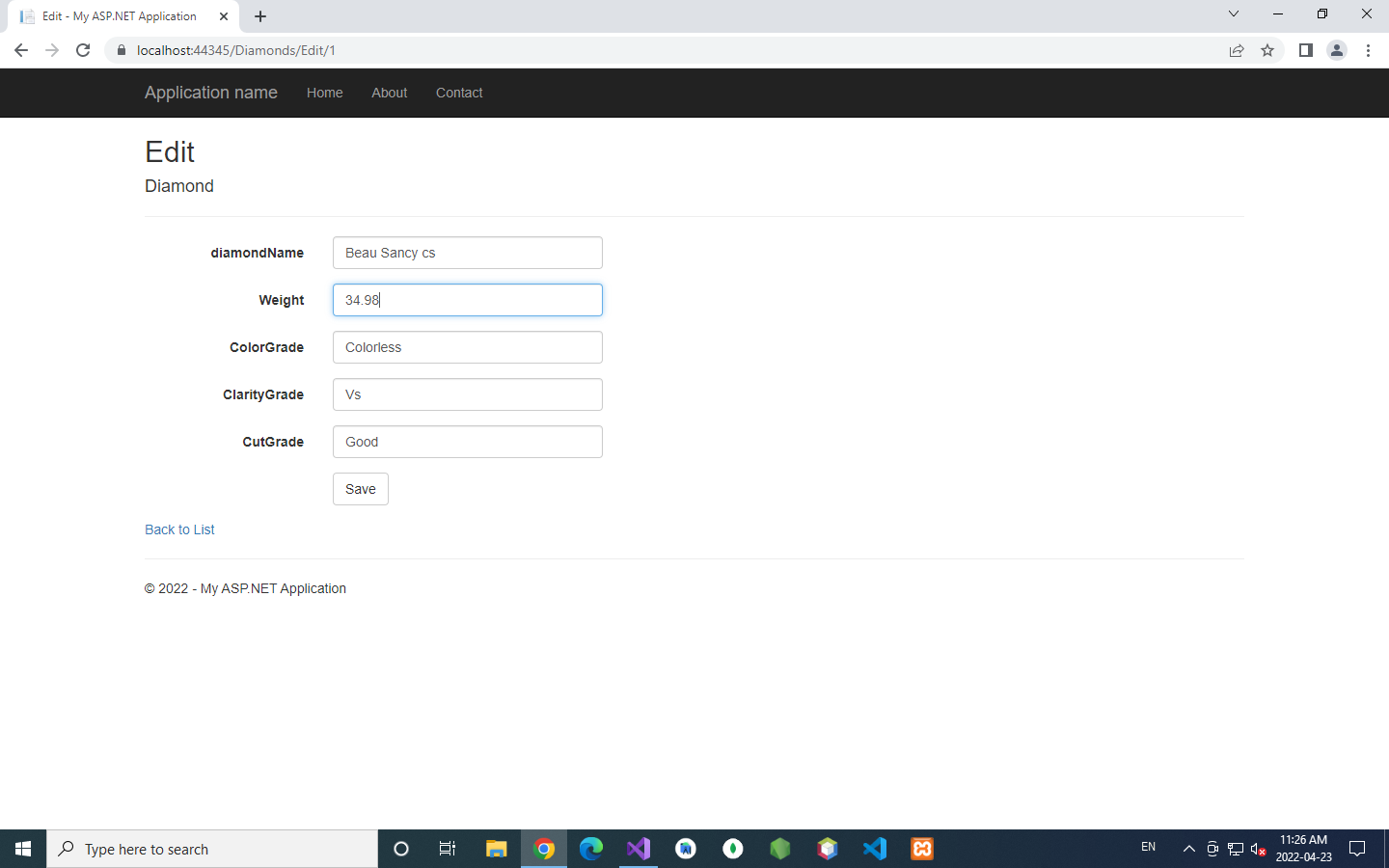
}

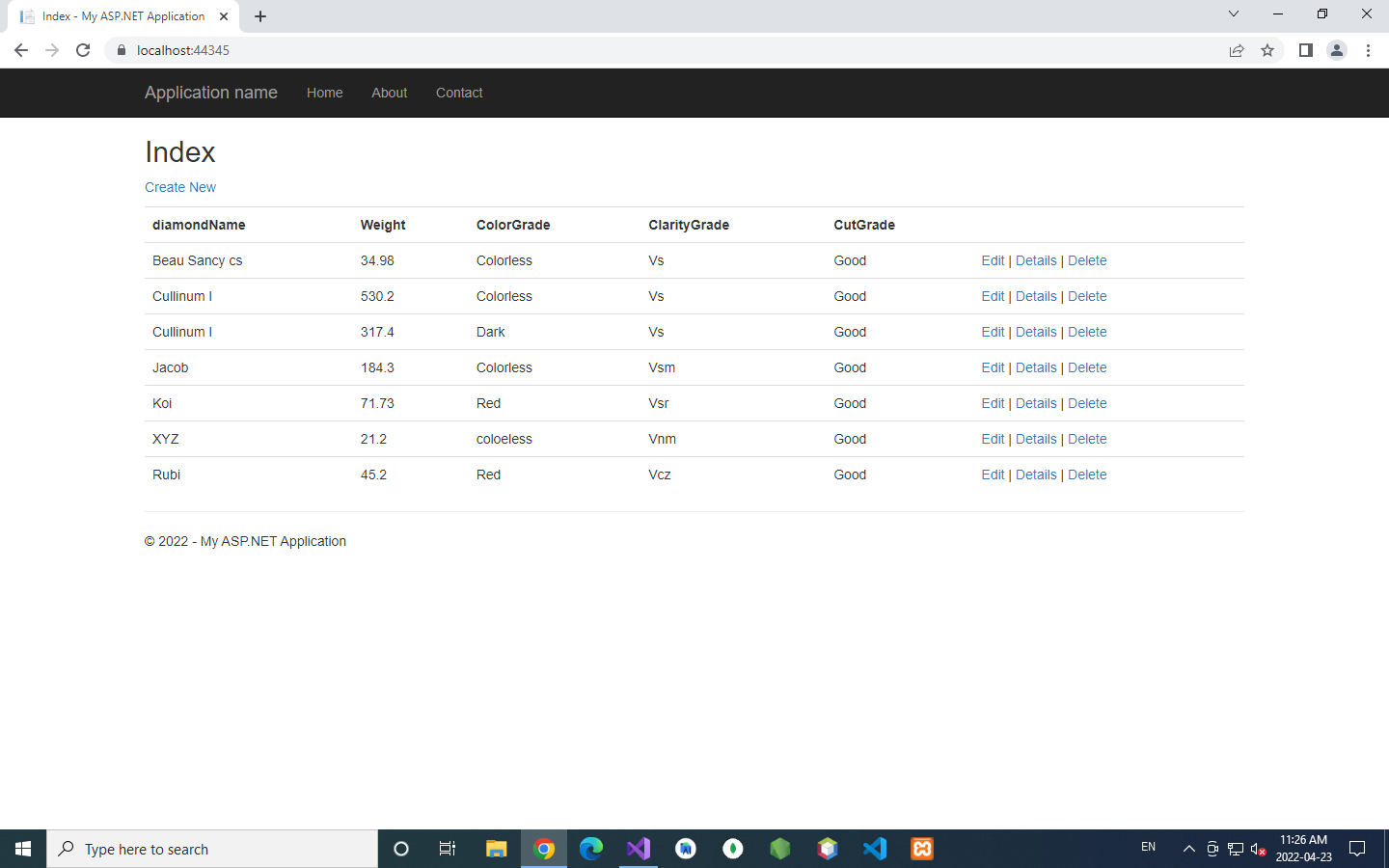
**OutPut:-**

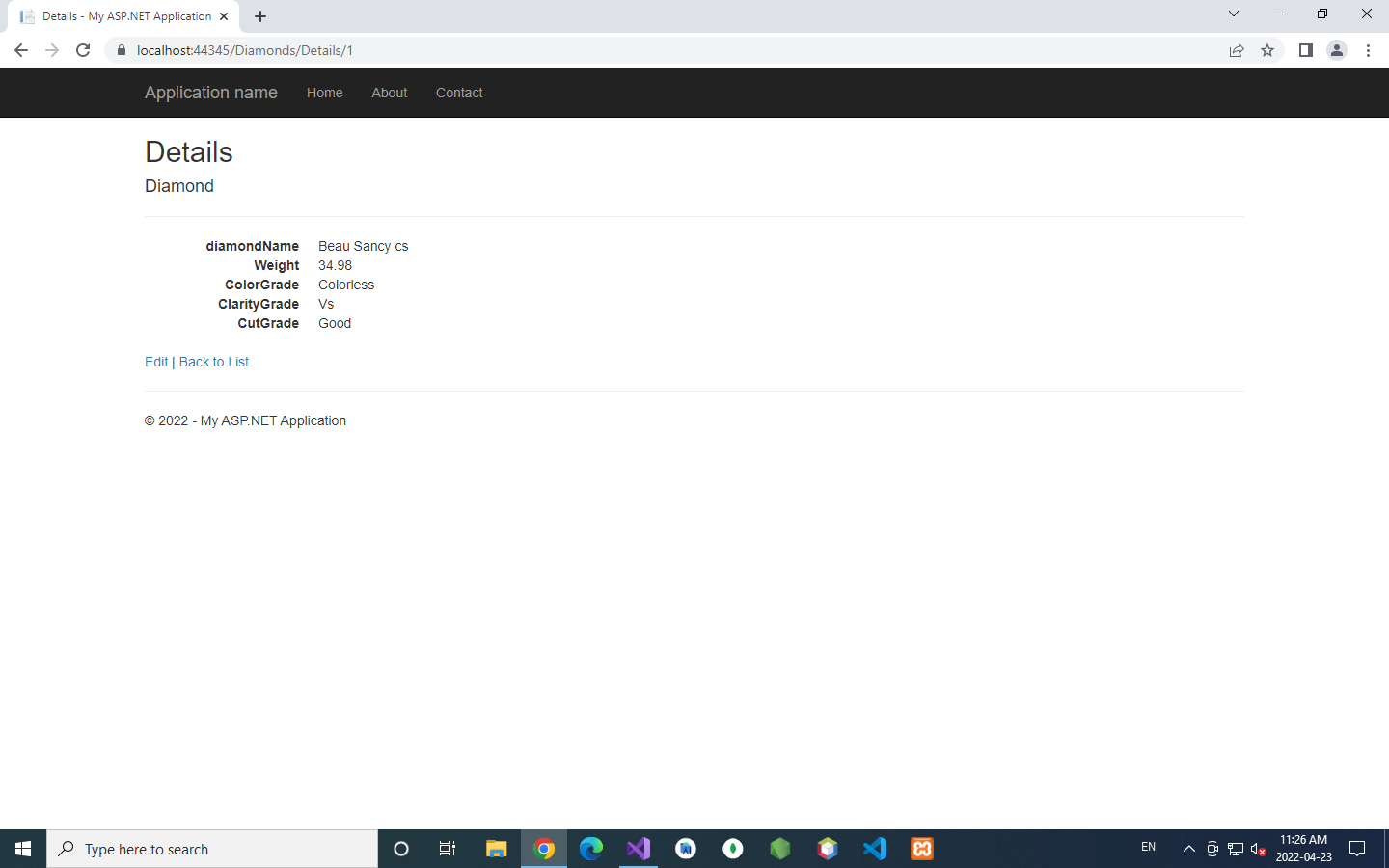
****

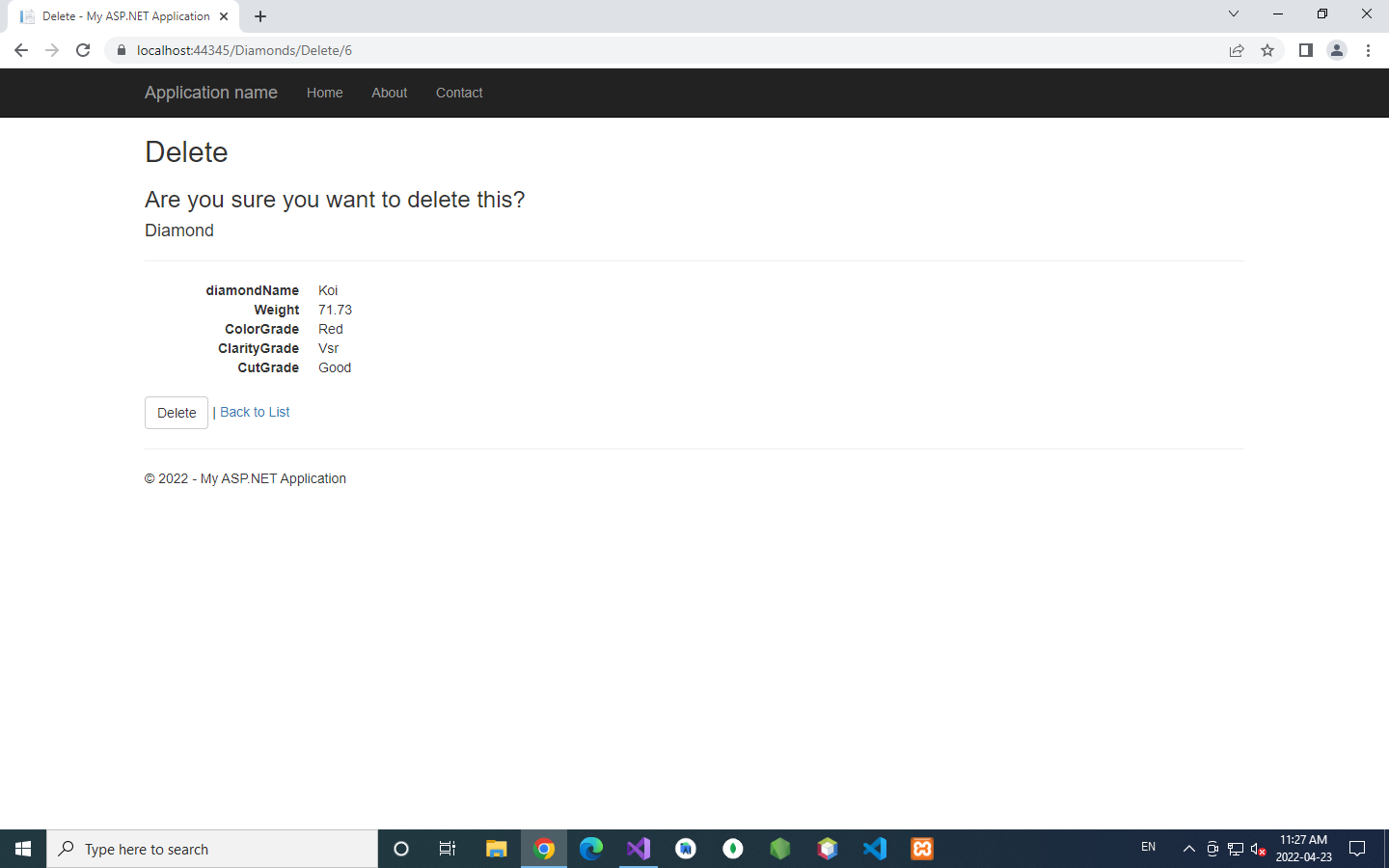
****

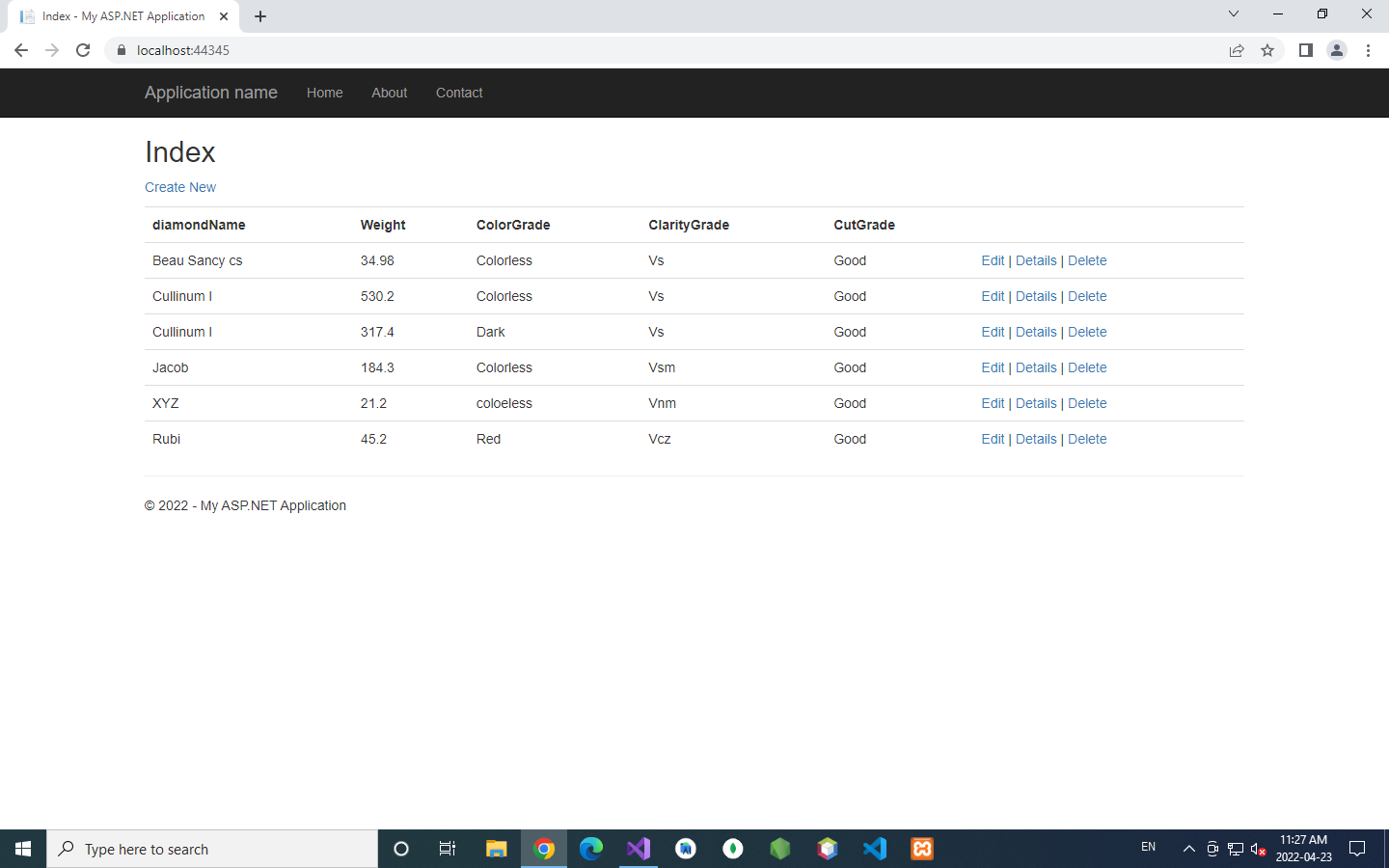
****

****

****

****

****

****