**Med Info Data Entry \_Source**

RouteConfig.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Routing;

namespace MedInfoDataEntry

{

public class RouteConfig

{

public static void RegisterRoutes(RouteCollection routes)

{

routes.IgnoreRoute("{resource}.axd/{\*pathInfo}");

routes.MapRoute(

name: "Default1",

url: "{controller}/{action}/{id}",

defaults: new { controller = "MedInfoes", action = "Index", id =

UrlParameter.Optional }

);

routes.MapRoute(

name: "Default",

url: "{controller}/{action}/{id}",

defaults: new { controller = "Home", action = "Index", id =

UrlParameter.Optional }

);

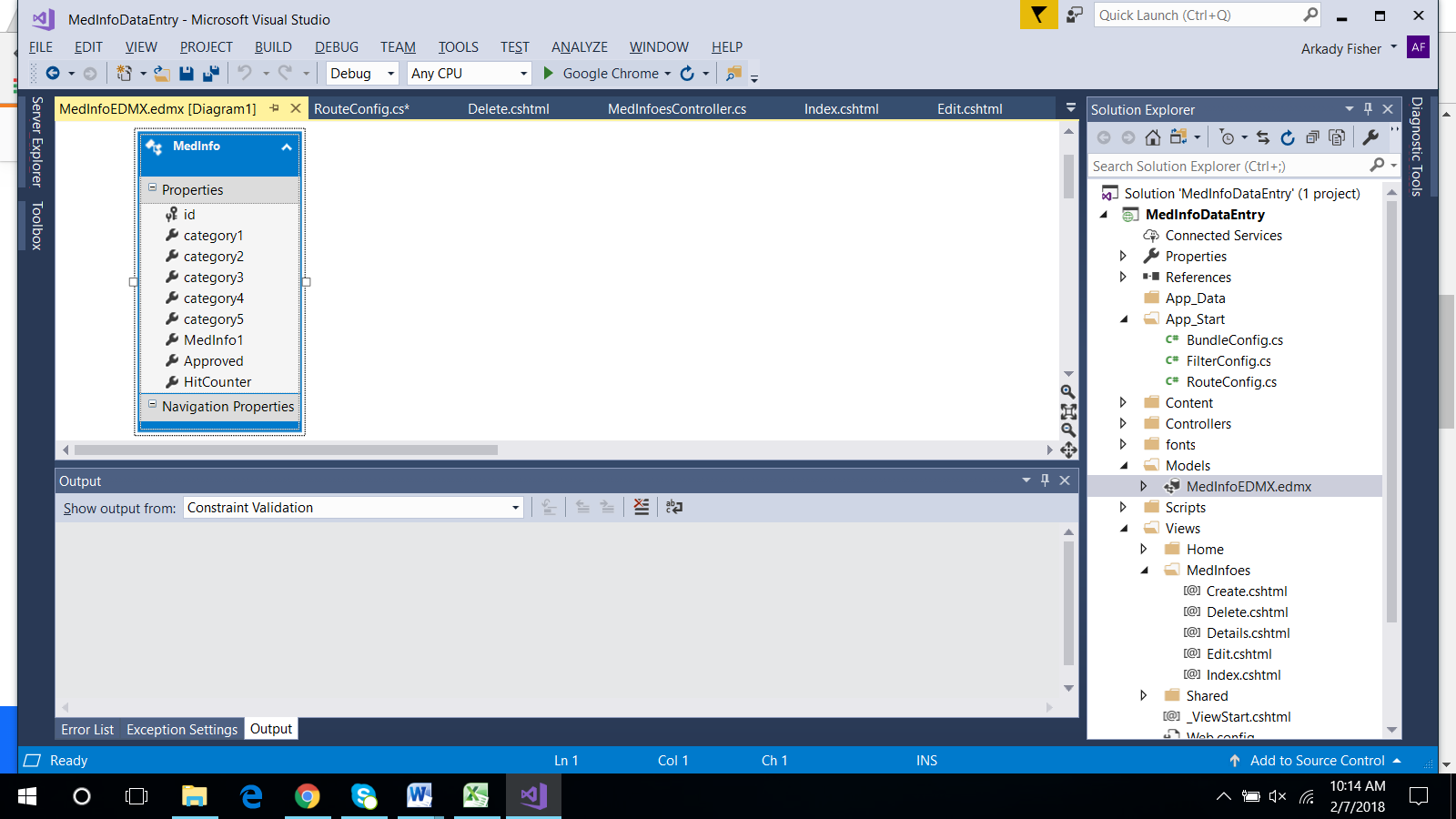
}

}

}

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

Models



Controllers

MedInfoesController.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 MedInfoDataEntry.Models;

namespace MedInfoDataEntry.Controllers

{

public class MedInfoesController : Controller

{

private MedInfoDBEntities db = new MedInfoDBEntities();

// GET: MedInfoes

public ActionResult Index(string currentFilter, string SearchString)

{

int id = Convert.ToInt32(SearchString);

ViewBag.CurrentFilter = SearchString;

var info = from inf in db.MedInfoes

select inf;

if (!String.IsNullOrEmpty(SearchString))

{

info = info.Where(i => i.id == id);

}

return View(info);

}

// GET: MedInfoes/Details/5

public ActionResult Details(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

MedInfo medInfo = db.MedInfoes.Find(id);

if (medInfo == null)

{

return HttpNotFound();

}

return View(medInfo);

}

// GET: MedInfoes/Create

public ActionResult Create()

{

return View();

}

// POST: MedInfoes/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 = "id,category1,category2,category3,category4,category5,MedInfo1,Approved,HitCounter")] MedInfo medInfo)

{

if (ModelState.IsValid)

{

db.MedInfoes.Add(medInfo);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(medInfo);

}

// GET: MedInfoes/Edit/5

public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

MedInfo medInfo = db.MedInfoes.Find(id);

if (medInfo == null)

{

return HttpNotFound();

}

return View(medInfo);

}

// POST: MedInfoes/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 = "id,category1,category2,category3,category4,category5,MedInfo1,Approved,HitCounter")] MedInfo medInfo)

{

if (ModelState.IsValid)

{

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

db.SaveChanges();

return RedirectToAction("Index");

}

return View(medInfo);

}

// GET: MedInfoes/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

MedInfo medInfo = db.MedInfoes.Find(id);

if (medInfo == null)

{

return HttpNotFound();

}

return View(medInfo);

}

// POST: MedInfoes/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

MedInfo medInfo = db.MedInfoes.Find(id);

db.MedInfoes.Remove(medInfo);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}

}

Views

MedInfoes

Index.cshtml

@model IEnumerable<MedInfoDataEntry.Models.MedInfo>

@{

ViewBag.Title = "Index";

}

<h2>Index</h2>

<p>

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

</p>

@using (Html.BeginForm("Index", "MedInfoes", FormMethod.Get))

{

<p>

Find by id: @Html.TextBox("SearchString", ViewBag.CurrentFilter as string)

<input type="submit" value="Search" />

</p>

}

<table class="table">

<tr>

<th>

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

</th>

<th>

@Html.DisplayNameFor(model => model.category1)

</th>

<th>

@Html.DisplayNameFor(model => model.category2)

</th>

<th>

@Html.DisplayNameFor(model => model.category3)

</th>

<th>

@Html.DisplayNameFor(model => model.category4)

</th>

<th>

@Html.DisplayNameFor(model => model.category5)

</th>

<th>

@Html.DisplayNameFor(model => model.MedInfo1)

</th>

<th>

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

</th>

<th>

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

</th>

<th></th>

</tr>

@foreach (var item in Model)

{

<tr>

<td>

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

</td>

<td>

@Html.DisplayFor(modelItem => item.category1)

</td>

<td>

@Html.DisplayFor(modelItem => item.category2)

</td>

<td>

@Html.DisplayFor(modelItem => item.category3)

</td>

<td>

@Html.DisplayFor(modelItem => item.category4)

</td>

<td>

@Html.DisplayFor(modelItem => item.category5)

</td>

<td>

@Html.DisplayTextFor(modelItem => item.MedInfo1)

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

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

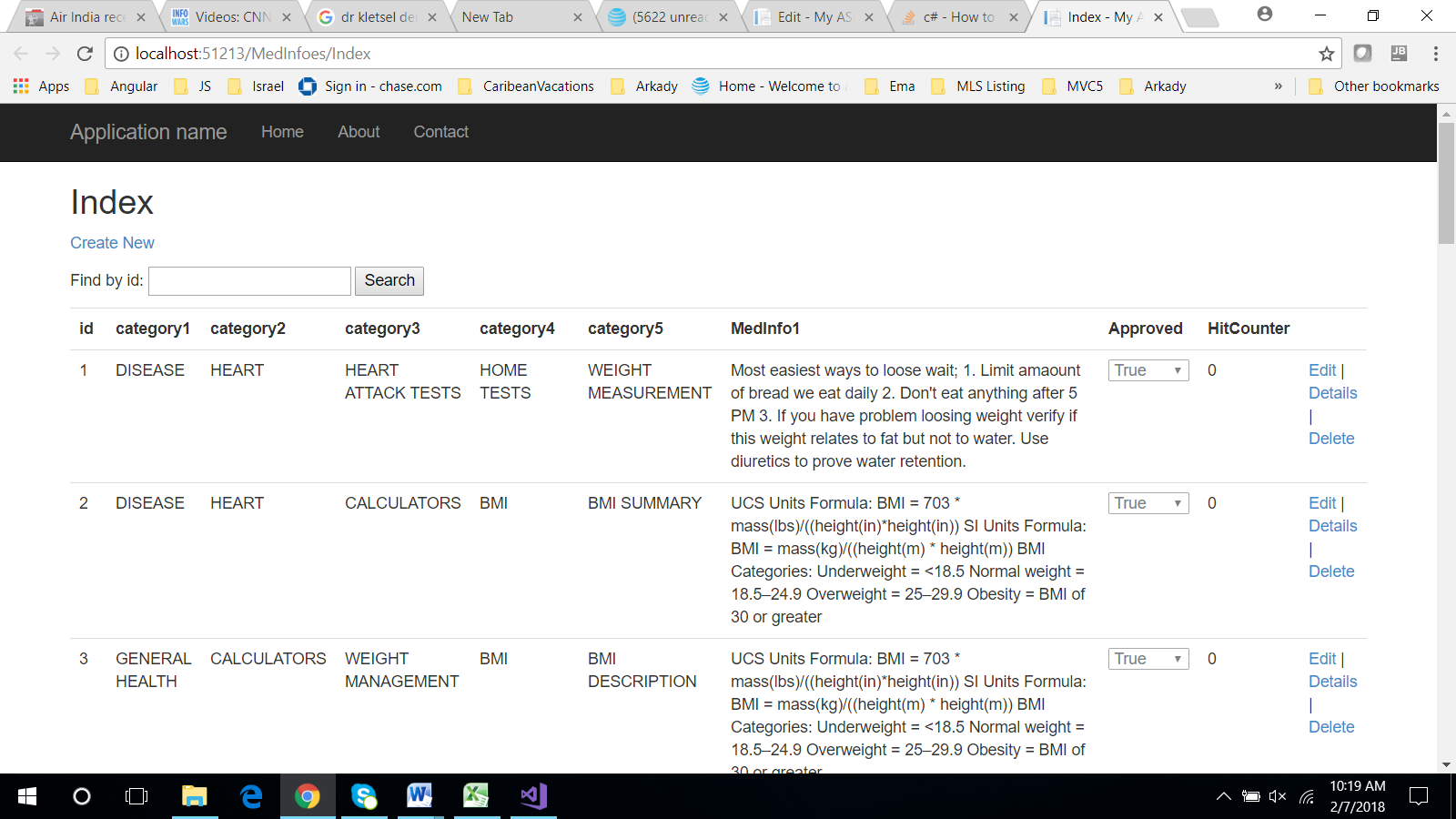
@Html.ActionLink("Delete", "Delete", new { id = item.id })

</td>

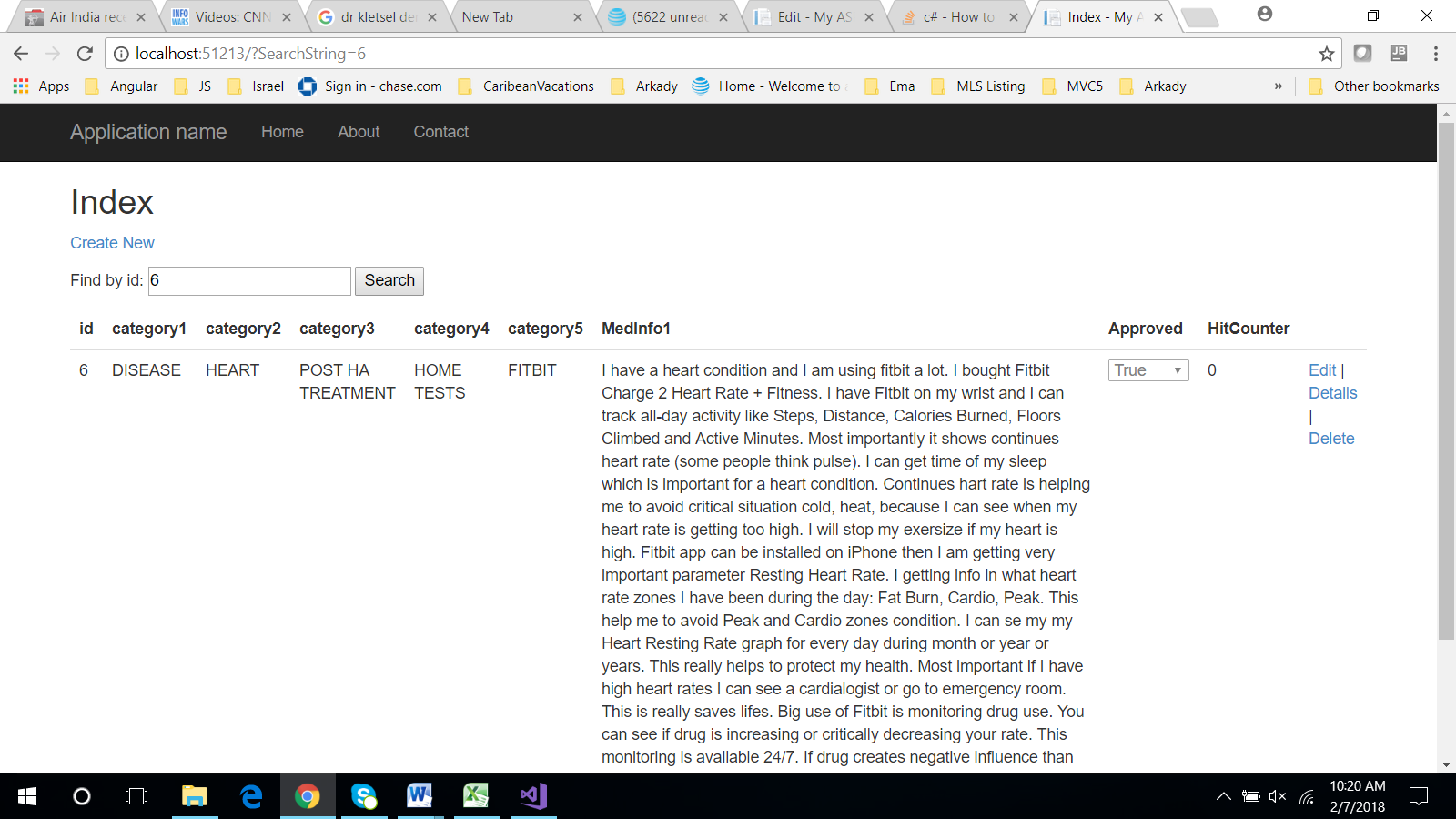
</tr>

}

</table>



Search



Edit.cshtml

@model MedInfoDataEntry.Models.MedInfo

@{

ViewBag.Title = "Edit";

}

<h2>Edit</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>MedInfo</h4>

<hr />

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

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

<div class="form-group">

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

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

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

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.category1, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.category1, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.category1, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.category2, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.category2, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.category2, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.category3, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.category3, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.category3, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.category4, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.category4, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.category4, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.category5, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.category5, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.category5, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

<div class="form-group">

@Html.LabelFor(model => model.MedInfo1, htmlAttributes: new { @class = "control-label col-md-2" })

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

@\*@Html.TextAreaFor(model => model.MedInfo1, new { htmlAttributes = new { @class = "form-control" } })\*@

@Html.TextAreaFor(model => model.MedInfo1, 10, 120, htmlAttributes: new { style = "width: 100%; max-width: 100%;" })

@Html.ValidationMessageFor(model => model.MedInfo1, "", new { @class = "text-danger" })

</div>

</div>

</div>

<div class="form-group">

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

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

<div class="checkbox">

@Html.EditorFor(model => model.Approved)

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

</div>

</div>

</div>

<div class="form-group">

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

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

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

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

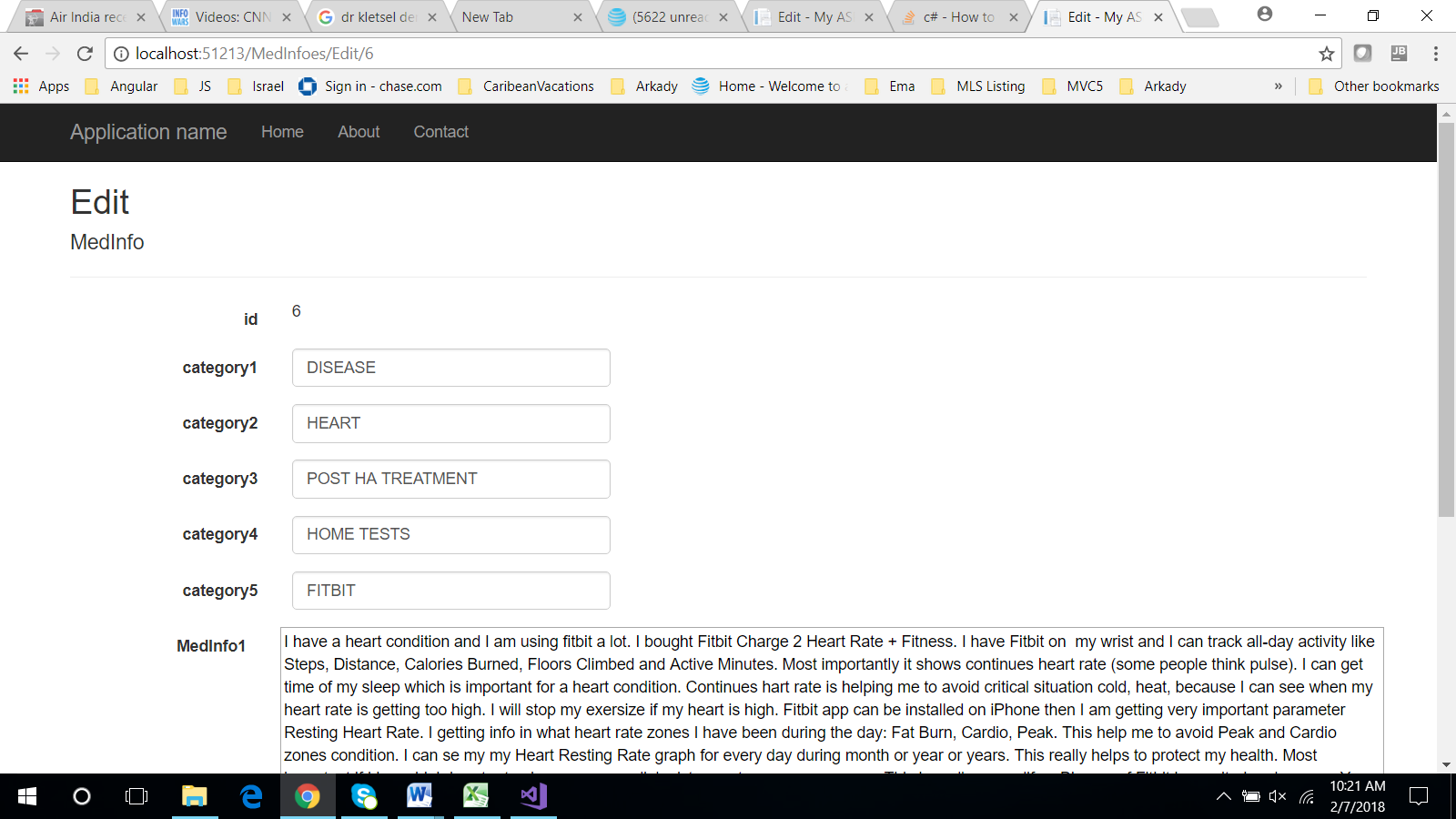
@Html.ActionLink("Back to List", "Index")

</div>

@section Scripts {

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

}



Details.cshtml

@model MedInfoDataEntry.Models.MedInfo

@{

ViewBag.Title = "Edit";

}

<h2>Edit</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>MedInfo</h4>

<hr />

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

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

<div class="form-group">

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

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

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

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.category1, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.category1, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.category1, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.category2, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.category2, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.category2, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.category3, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.category3, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.category3, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.category4, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.category4, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.category4, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.category5, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.category5, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.category5, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

<div class="form-group">

@Html.LabelFor(model => model.MedInfo1, htmlAttributes: new { @class = "control-label col-md-2" })

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

@\*@Html.TextAreaFor(model => model.MedInfo1, new { htmlAttributes = new { @class = "form-control" } })\*@

@Html.TextAreaFor(model => model.MedInfo1, 10, 120, htmlAttributes: new { style = "width: 100%; max-width: 100%;" })

@Html.ValidationMessageFor(model => model.MedInfo1, "", new { @class = "text-danger" })

</div>

</div>

</div>

<div class="form-group">

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

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

<div class="checkbox">

@Html.EditorFor(model => model.Approved)

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

</div>

</div>

</div>

<div class="form-group">

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

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

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

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

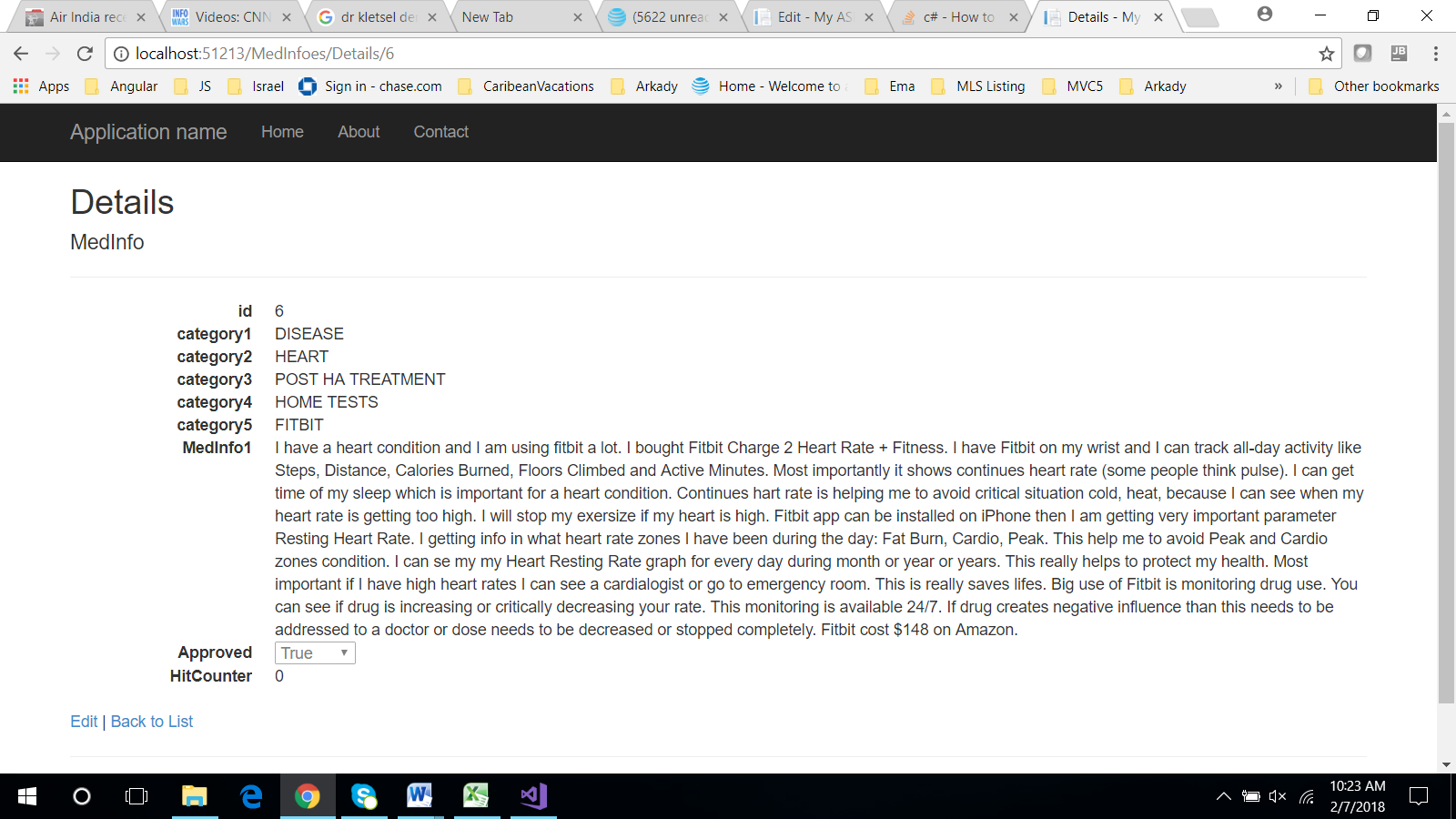
@Html.ActionLink("Back to List", "Index")

</div>

@section Scripts {

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

}



Delete.cshtml

@{

ViewBag.Title = "Delete";

}

<h2>Delete</h2>

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

<div>

<h4>MedInfo</h4>

<hr />

<dl class="dl-horizontal">

<dt>

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

</dt>

<dd>

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

</dd>

<dt>

@Html.DisplayNameFor(model => model.category1)

</dt>

<dd>

@Html.DisplayFor(model => model.category1)

</dd>

<dt>

@Html.DisplayNameFor(model => model.category2)

</dt>

<dd>

@Html.DisplayFor(model => model.category2)

</dd>

<dt>

@Html.DisplayNameFor(model => model.category3)

</dt>

<dd>

@Html.DisplayFor(model => model.category3)

</dd>

<dt>

@Html.DisplayNameFor(model => model.category4)

</dt>

<dd>

@Html.DisplayFor(model => model.category4)

</dd>

<dt>

@Html.DisplayNameFor(model => model.category5)

</dt>

<dd>

@Html.DisplayFor(model => model.category5)

</dd>

<dt>

@Html.DisplayNameFor(model => model.MedInfo1)

</dt>

<dd>

@Html.DisplayFor(model => model.MedInfo1)

</dd>

<dt>

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

</dt>

<dd>

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

</dd>

<dt>

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

</dt>

<dd>

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

</dd>

</dl>

@using (Html.BeginForm()) {

@Html.AntiForgeryToken()

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

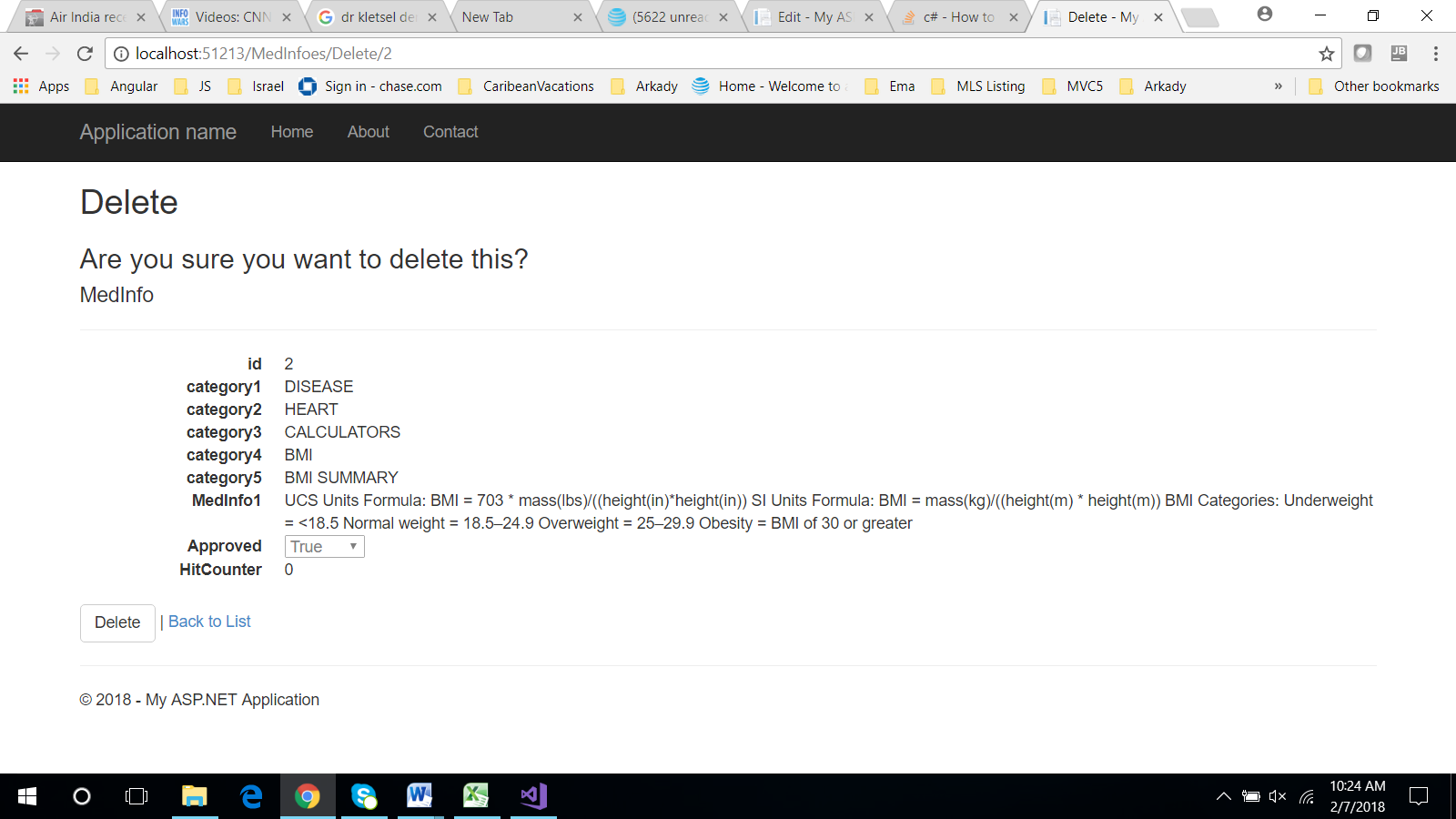
<input type="submit" value="Delete" class="btn btn-default" /> |

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

</div>

}

</div>



Create.cshtml

@model MedInfoDataEntry.Models.MedInfo

@{

ViewBag.Title = "Create";

}

<h2>Create</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>MedInfo</h4>

<hr />

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

<div class="form-group">

@Html.LabelFor(model => model.category1, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.category1, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.category1, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.category2, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.category2, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.category2, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.category3, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.category3, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.category3, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.category4, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.category4, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.category4, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.category5, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.category5, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.category5, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.MedInfo1, htmlAttributes: new { @class = "control-label col-md-2" })

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

@\*@Html.EditorFor(model => model.MedInfo1, new { htmlAttributes = new { @class = "form-control" } })\*@

@Html.TextAreaFor(model => model.MedInfo1, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.MedInfo1, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

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

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

<div class="checkbox">

@Html.EditorFor(model => model.Approved)

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

</div>

</div>

</div>

<div class="form-group">

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

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

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

@Html.ValidationMessageFor(model => model.HitCounter, "", 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")

}

