**1. Create a New Project With Project Name : ProjectName**

**2. Create DB : DB\_Name (Create Table )**

CREATE TABLE [dbo].[station] (

[stationid] INT IDENTITY (1, 1) NOT NULL,

[stationname] NVARCHAR (50) NULL,

PRIMARY KEY CLUSTERED ([stationid] ASC)

);

CREATE TABLE [dbo].[train] (

[trainid] INT IDENTITY (1, 1) NOT NULL,

[trainname] NVARCHAR (50) NULL,

[capacity] INT NULL,

[isactive] BIT NULL,

[startdate] DATE NULL,

[picture] NVARCHAR (MAX) NULL,

PRIMARY KEY CLUSTERED ([trainid] ASC)

);

CREATE TABLE [dbo].[route] (

[routeid] INT IDENTITY (1, 1) NOT NULL,

[trainid] INT NULL,

[stationid] INT NULL,

PRIMARY KEY CLUSTERED ([routeid] ASC),

CONSTRAINT [FK\_Table\_ToTable] FOREIGN KEY ([trainid]) REFERENCES [dbo].[train] ([trainid]),

CONSTRAINT [FK\_route\_ToTable] FOREIGN KEY ([stationid]) REFERENCES [dbo].[station] ([stationid])

);

**3.Model Biend**

**4.Create a Folder in side the Model folder “ViewModel” and create a class**

**TrainViewModel.cs** and past :

public TrainViewModel()

{

this.routes = new List<route>();

}

public int trainid { get; set; }

[Display(Name = "Train Name")]

public string trainname { get; set; }

[Required(ErrorMessage = "Train Name is Required")]

[Display(Name = "Capacity")]

public Nullable<int> capacity { get; set; }

[Required(ErrorMessage = "Train Capacity is Required")]

[Display(Name = "Is Active")]

public bool isactive { get; set; }

[Display(Name = "Start Date")]

public Nullable<System.DateTime> startdate { get; set; }

[Display(Name = "Picture")]

public string picture { get; set; }

public HttpPostedFileBase picturefile { get; set; }

public List<route> routes { get; set; }

**5. Maek Controller**

public ActionResult Index()

{

var train = db.trains.Include(r => r.routes.Select(s => s.station)).OrderBy(t => t.trainid).ToList();

return View(train);

}

[HttpGet]

public ActionResult Create()

{

return View();

}

[HttpPost]

public ActionResult Create(TrainViewModel trainVm, int[] stationid)

{

if (ModelState.IsValid)

{

var train = new train()

{

trainname = trainVm.trainname,

capacity = trainVm.capacity,

isactive = trainVm.isactive,

startdate = trainVm.startdate,

};

HttpPostedFileBase file = trainVm.picturefile;

if (file != null)

{

string fileName = Path.Combine("/Images/", DateTime.Now.Ticks.ToString() + Path.GetExtension(file.FileName));

file.SaveAs(Server.MapPath(fileName));

train.picture = fileName;

}

foreach (var stat in stationid)

{

var s = db.stations.FirstOrDefault(st => st.stationid == stat);

if (s != null)

{

var route = new route()

{

train = train,

stationid = s.stationid,

trainid = train.trainid,

};

db.routes.Add(route);

}

}

db.SaveChanges();

return RedirectToAction("Index");

}

return View(trainVm);

}

public ActionResult AddStation(int? id)

{

ViewBag.station = new SelectList(db.stations.ToList(), "stationid", "stationname", (id != null) ? id.ToString() : "");

return PartialView("\_AddStation");

}

public ActionResult Edit(int? id)

{

if (id != null)

{

var train = db.trains.Find(id);

if (train != null)

{

var newTrian = new TrainViewModel()

{

trainname = train.trainname,

capacity = train.capacity,

isactive = train.isactive,

picture = train.picture,

startdate = train.startdate,

trainid = train.trainid,

};

var listRoute = db.routes.Where(x => x.trainid == train.trainid).ToList();

newTrian.routes = listRoute;

return View(newTrian);

}

}

return HttpNotFound();

}

[HttpPost]

public ActionResult Edit(TrainViewModel trainVm, int[] stationid)

{

if (ModelState.IsValid)

{

var train = db.trains.Find(trainVm.trainid);

if (train == null)

{

return HttpNotFound();

}

train.trainname = trainVm.trainname;

train.capacity = trainVm.capacity;

train.isactive = trainVm.isactive;

train.startdate = trainVm.startdate;

HttpPostedFileBase file = trainVm.picturefile;

if (file != null)

{

string fileName = Path.Combine("/Images/", DateTime.Now.Ticks.ToString() + Path.GetExtension(file.FileName));

file.SaveAs(Server.MapPath(fileName));

train.picture = fileName;

}

else

{

train.picture = trainVm.picture;

}

var rs = db.routes.Where(t => t.trainid == train.trainid).ToList();

foreach (var t in rs)

{

db.routes.Remove(t);

}

foreach (var stat in stationid)

{

var s = db.stations.FirstOrDefault(st => st.stationid == stat);

if (s != null)

{

var route = new route()

{

train = train,

stationid = s.stationid,

trainid = train.trainid,

};

db.routes.Add(route);

}

}

db.SaveChanges();

return RedirectToAction("Index");

}

return View(trainVm);

}

public ActionResult Delete(int? id)

{

var train = db.trains.Find(id);

var sta = db.routes.Where(t => t.trainid == train.trainid).ToList();

if (sta != null)

{

foreach (var t in sta)

{

db.routes.Remove(t);

}

}

db.trains.Remove(train);

db.SaveChanges();

return RedirectToAction("Index");

}

**6.make view pages Index:**

@model IEnumerable<TrainApp.Models.train>

@{

ViewBag.Title = "Index";

}

<h2>Index</h2>

<section>

<div>

@Html.ActionLink("Create", "Create", null, new { @class = "btn btn-primary" })

</div>

<table class="table text-center border-1">

<thead>

<tr>

<th>

@Html.DisplayNameFor(m => m.trainid)

</th>

<th>

@Html.DisplayNameFor(m => m.trainname)

</th>

<th>

@Html.DisplayNameFor(m => m.startdate)

</th>

<th>

@Html.DisplayNameFor(m => m.isactive)

</th>

<th>

@Html.DisplayNameFor(m => m.capacity)

</th>

<th>

@Html.DisplayNameFor(m => m.picture)

</th>

</tr>

</thead>

<tbody>

@foreach (var t in Model)

{

<tr>

<td>

@Html.DisplayFor(m => t.trainid)

</td>

<td>

@Html.DisplayFor(m => t.trainname)

</td>

<td>

@Html.DisplayFor(m => t.startdate)

</td>

<td>

@Html.DisplayFor(m => t.isactive)

</td>

<td>

@Html.DisplayFor(m => t.capacity)

</td>

<td>

<img src="@t.picture" height="80" width="80" />

</td>

</tr>

<tr>

<td colspan="6">

<table class="table text-center table-active">

<thead>

<tr>

<th>

@Html.Label("station id")

</th>

<th>

@Html.Label("station Naem")

</th>

</tr>

</thead>

<tbody>

@foreach (var s in t.routes)

{

<tr>

<td>

@Html.DisplayFor(m => s.station.stationid)

</td>

<td>

@Html.DisplayFor(m => s.station.stationname)

</td>

</tr>

}

</tbody>

</table>

</td>

</tr>

<tr>

<td colspan="12">

@Html.ActionLink("Edit", "Edit", new { id = t.trainid }, new { @class = "btn btn-info" })

@Html.ActionLink("Delete", "Delete", new { id = t.trainid }, new { @class = "btn btn-danger" })

</td>

</tr>

}

</tbody>

</table>

</section>

**7. view Create**

@model TrainApp.Models.ViewModel.TrainViewModel

@{

ViewBag.Title = "Create";

}

@using (Html.BeginForm("Create", "Train", FormMethod.Post, new { enctype = "multipart/form-data" }))

{

<div class="form-horizontal">

<h4>Train</h4>

<hr />

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

<div class="form-group">

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

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

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

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

</div>

</div>

<div class="form-group">

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

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

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

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

</div>

</div>

<div class="form-group">

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

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

<div class="checkbox">

@Html.CheckBoxFor(model => model.isactive)

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

</div>

</div>

</div>

<div class="form-group">

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

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

@Html.TextBoxFor(model => model.startdate, new { htmlAttributes = new { @class = "form-control" }, Type = "date" })

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

</div>

</div>

<div class="form-group">

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

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

@Html.TextBoxFor(model => model.picturefile, new { htmlAttributes = new { @class = "form-control" }, Type = "file" })

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

</div>

</div>

<div>

<div>

@Html.ActionLink("add more", "", null, new { id = "addmore" })

</div>

<div id="con">

@Html.Action("AddStation", "Train")

</div>

</div>

<div class="form-group">

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

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

</div>

</div>

</div>

}

<div>

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

</div>

**8. View Edit**

@model TrainApp.Models.ViewModel.TrainViewModel

@{

ViewBag.Title = "Edit";

}

@using (Html.BeginForm("Edit", "Train", FormMethod.Post, new { enctype = "multipart/form-data" }))

{

<div class="form-horizontal">

<h4>train</h4>

<hr />

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

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

<div class="form-horizontal">

<h4>train</h4>

<hr />

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

<div class="form-group">

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

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

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

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

</div>

</div>

<div class="form-group">

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

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

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

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

</div>

</div>

<div class="form-group">

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

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

<div class="checkbox">

@Html.CheckBoxFor(model => model.isactive)

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

</div>

</div>

</div>

<div class="form-group">

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

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

@\*@Html.TextBoxFor(model => model.startdate, new { htmlAttributes = new { @class = "form-control" }, Type = "date" })

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

@Html.TextBox("parchacedate", Model.startdate?.ToString("yyyy-MM-dd"), new { @class = "form-control", type = "date" })

</div>

</div>

@\*<div class="form-group">

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

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

@Html.TextBoxFor(model => model.picturefile, new { htmlAttributes = new { @class = "form-control" }, Type = "file" })

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

</div>

</div>\*@

<div class="form-group">

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

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

<img src="@Model.picture" alt="Student Picture" width="100" height="100" />

@Html.TextBoxFor(model => model.picturefile, new { @class = "form-control", type = "file" })

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

</div>

</div>

<div>

<div>

@Html.ActionLink("add more", "", null, new { id = "addmore" })

</div>

<div id="con">

@foreach (var a in Model.routes)

{

@Html.Action("AddStation", "Train", new { id = a.stationid })

}

</div>

</div>

<div class="form-group">

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

<input type="submit" value="Update" class="btn btn-success" />

</div>

</div>

</div>

</div>

}

<div>

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

</div>

@\*<script>

$(document).ready(() => {

$("#addmore").click((e) => {

e.preventDefault();

$.ajax({

url: "/Train/AddStation",

type: "get",

success: (d) => {

$("#con").append(d);

}

});

});

</script>\*@

**9. \_Layout.cshtml**

<Script>

$(document).ready(() => {

$("#addmore").click((e) => {

e.preventDefault();

$.ajax({

url: "/Train/AddStation",

type: "get",

success: (d) => {

$("#con").append(d);

}

});

});

$(document).on("click", "#remove", (e) => {

e.preventDefault();

$(e.currentTarget).closest("#rc").remove();

})

});

</Script>

10. Create partial view

@model TrainApp.Models.station

<div id="rc">

<div>

@Html.DropDownListFor(s => s.stationid, ViewBag.station as SelectList, "--select--")

</div>

<div>

@Html.ActionLink("remove", "", null, new { id = "remove" })

</div>

</div>