1. File>New>Project>Web Tab>ASP>Net Web Application(>Net Core)>OK>Web Application>OK
2. **Create database mydb and create tables as:**

create table items

(

itemcode varchar(100) primary key,

itemname varchar(100),

picture varchar(200)

)

Create table student

(

id int primary key,

name varchar(100),

fee money,

picture nvarchar(500)

)

create table Fee

(

voucherno nvarchar(100) primary key,

inputDate datetime,

studentId int foreign key references student(id),

headname nvarchar(100),

amount money

)

create table examresult

(

examsl int,

studentid int foreign key references student(id),

examname varchar(100),

institution varchar(200),

board varchar(200),

result varchar(100),

primary key(examsl,studentid)

)

1. **Tools –> NuGet Package Manager –> Package Manager Console>Write code:**

Scaffold-DbContext "Server=(localdb)\mssqllocaldb;Database=mydb;Trusted\_Connection=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Model/Db

1. Items.cs,student.cs, fees.cs, ExamResult.cs and mydbcontext.cs will be created in Model/Db
2. Right Click Controller>Add Controller>MVC Controller with views, entity framework>Add>Model Class: items(..), Data context class: mydbcontext(…)>Add
3. Write code in startup.cs as

public void ConfigureServices(IServiceCollection services)

{

services.AddMvc();

**services.AddDbContext<mydbContext>();**

}

1. Browse Main Page>……………………/items and enjoy
2. Add action in ItemsController.cs

public ActionResult UploadFile(string id)

{

return View(\_context.Items.Find(id));

}

[HttpPost, ActionName("UploadFile")]

public ActionResult UploadFile(IFormFile file, string id)

{

try

{

if (file == null || file.Length == 0)

return Content("file not selected");

if (file.Length > 0)

{

var path = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot","images/"+ file.FileName);

using (var stream = new FileStream(path, FileMode.Create))

{

file.CopyTo(stream);

}

Items st = (from s in \_context.Items where s.Itemcode == id select s).First();

st.Picture = file.FileName;

\_context.SaveChanges();

}

ViewBag.Message = "File Uploaded Successfully!!" + ":" + id;

return RedirectToAction("Index");

}

catch

{

ViewBag.Message = "File upload failed!!";

return View();

}

}

1. Build>Rebuild>Close all files>Open ItemsController.cs from solution explorer>Right click Index>Add View>Template: Details>Model:Items(…)>Select Layout>Add
2. Run The Project and check whether UploadFile works or not.( <http://localhost:49803/Items/UploadFile/I-001>)
3. Add the following lines in UploadFile.cshtml

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

{

@Html.TextBox("file", "", new { type = "file" }) <br />

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

}

1. Replace @Html.DisplayFor(modelItem => item.Picture) by

@if (item.Picture != null && System.IO.File.Exists(System.IO.Path.Combine(System.IO.Directory.GetCurrentDirectory(), "wwwroot", "images\\" + item.Picture)))

{

<img src="~/Images/@Html.DisplayFor(modelItem => item.Picture)" alt="@item.Itemname"

width="100" class="img-responsive" />

<br />

@Html.ActionLink("Change Picture", "UploadFile", new { id = item.Itemcode })

}

else

{

<img src="~/Images/cross.jpg" alt="No Image" width="100" class="img-responsive" />

<br />

@Html.ActionLink("Upload Picture", "UploadFile", new { id = item.Itemcode })

}

1. Test by using Items/index and enjoy
2. Controller>Right Click>Add Controller>MVC Controller with Views, using Entity Framework>Model Class: Student(...), DataContext Class: mydbEntities>Add
3. Repeat step 14 for fee and examresult.
4. There is problem in auto crud operation on ExamResults. Because, primary key is formed from two fields(examsl and studentid). Hence index view and ExamResultsController should changed.
5. Chang last three lines in views/ExamResults/index.cshtml

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

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

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

1. Chage ExamResultsController.cs as:

public class ExamresultsController : Controller

{

private readonly mydbContext \_context;

public ExamresultsController(mydbContext context)

{

\_context = context;

}

public async Task<IActionResult> Index()

{

var mydbContext = \_context.Examresult.Include(e => e.Student);

return View(await mydbContext.ToListAsync());

}

public async Task<IActionResult> Details(**int? examsl,int? studentid**)

{

if (**examsl == null || studentid==null**)

{

return NotFound();

}

var examresult = await \_context.Examresult

.Include(e => e.Student)

.SingleOrDefaultAsync(**m => m.Examsl == examsl && m.Studentid==studentid**);

if (examresult == null)

{

return NotFound();

}

return View(examresult);

}

public IActionResult Create()

{

ViewData["Studentid"] = new SelectList(\_context.Student, "Id", "Id");

return View();

}

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Create([Bind("Examsl,Studentid,Examname,Institution,Board,Result")] Examresult examresult)

{

if (ModelState.IsValid)

{

\_context.Add(examresult);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

ViewData["Studentid"] = new SelectList(\_context.Student, "Id", "Id", examresult.Studentid);

return View(examresult);

}

public async Task<IActionResult> Edit(**int? examsl,int? studentid**)

{

if (**examsl == null || studentid == null**)

{

return NotFound();

}

var examresult = await \_context.Examresult.SingleOrDefaultAsync(**m => m.Examsl == examsl && m.Studentid == studentid**);

if (examresult == null)

{

return NotFound();

}

ViewData["Studentid"] = new SelectList(\_context.Student, "Id", "Id", examresult.Studentid);

return View(examresult);

}

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Edit**([Bind("Examsl,Studentid,Examname,Institution,Board,Result")] Examresult examresult**)

{

if (ModelState.IsValid)

{

try

{

\_context.Update(examresult);

await \_context.SaveChangesAsync();

}

catch (DbUpdateConcurrencyException)

{

if (!ExamresultExists(examresult.Examsl,examresult.Studentid))

{

return NotFound();

}

else

{

throw;

}

}

return RedirectToAction(nameof(Index));

}

ViewData["Studentid"] = new SelectList(\_context.Student, "Id", "Id", examresult.Studentid);

return View(examresult);

}

public async Task<IActionResult> Delete(**int? examsl, int? studentid**)

{

if (**examsl == null || studentid == null**)

{

return NotFound();

}

var examresult = await \_context.Examresult

.Include(e => e.Student)

.SingleOrDefaultAsync(**m => m.Examsl == examsl && m.Studentid == studentid**);

if (examresult == null)

{

return NotFound();

}

return View(examresult);

}

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public async Task<IActionResult> DeleteConfirmed(**int? examsl,int? studentid**)

{

var examresult = await \_context.Examresult.SingleOrDefaultAsync(**m => m.Examsl == examsl && m.Studentid == studentid**);

\_context.Examresult.Remove(examresult);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

private bool ExamresultExists(**int examsl, int studentid**)

{

return \_context.Examresult.Any(**e => e.Examsl == examsl && e.Studentid == studentid**);

}

}

1. In Views/Shared/\_Layout.cshtml write the following lines after <li><**a** **asp-area**="" **asp-controller**="Home" **asp-action**="Contact">Contact</**a**></li>

<li><**a** **asp-area**="" **asp-controller**="Items" **asp-action**="Index">Items</**a**></li>

<li><**a** **asp-area**="" **asp-controller**="Students" **asp-action**="Index">Students</**a**></li>

<li><**a** **asp-area**="" **asp-controller**="Fees" **asp-action**="Index">Fees</**a**></li>

<li><**a** **asp-area**="" **asp-controller**="ExamResults" **asp-action**="Index">Exam Result</**a**></li>

<li><**a** **asp-area**="" **asp-controller**="Search" **asp-action**="StudentExamFee">Search</**a**></li>

1. Write two catch in Create action of Fees,Students, ExamResults and Items Controllers.

catch(DbUpdateException dbx)

{

ViewData["err"] = "Duplicate Voucher No";

return View();

}

catch(Exception ex)

{

ViewData["err"] = ex.Message.ToString();

return View();

}

1. Create a controller named SearchController(Empty Controller). Write the following codes there as follows:

private mydbContext db = new mydbContext();

public ActionResult StudentExamFee()

{

ViewBag.studentid = new SelectList(db.Student, "id", "name");

ViewBag.students = new List<Student>();

ViewBag.results = new List<Examresult>();

ViewBag.fees = new List<Fee>();

return View();

}

[HttpPost, ActionName("StudentExamFee")]

public ActionResult StudentExamFee(int studentid = 0)

{

try

{

ViewBag.studentid = new SelectList(db.Student, "id", "name");

ViewBag.students = (from st in db.Student select new { st.Id, st.Name });

List<Student> studentList = (from st in db.Student where st.Id == studentid select st).ToList();

ViewBag.students = studentList;

List<Examresult> res = (from st in db.Examresult where st.Studentid == studentid orderby st.Examsl select st).ToList();

ViewBag.results = res;

//List<Fee> fees = (from st in db.Fees where st.StudentId == studentid orderby st.voucherno select st).ToList();

//ViewBag.fees = fees;

List<Fee> fees = (from st in db.Fee join s in db.Student on st.StudentId equals s.Id where st.StudentId == studentid orderby st.Voucherno select st).ToList();

ViewBag.fees = fees;

ViewBag.Message = "Post Worked";

return View();

}

catch

{

ViewBag.Message = "File upload failed!!";

return View();

}

}

1. Right click SearchController in code>Add Action> StudentExamFee.cshtml(Empty)>Add
2. Write code in Views/Search/ StudentExamFee.cshtml as;

@model List<CoreWebApp\_2018\_08\_16.Model.Db.Student>

@{

ViewBag.Title = "StudentExamFee";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>Search Student, Exam Results and Fees Information</h2>

<h2>@ViewBag.Message</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>examresult</h4>

<hr />

<div class="form-group">

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

Student

</div>

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

<input class="form-control" type="text" id="studentid" name="studentid" placeholder="Your id goes here" />

</div>

</div>

<div class="form-group">

<div class="col-md-12">

<input type="submit" value="Show" class="btn btn-primary form-control" />

</div>

</div>

</div>

}

<hr />

@foreach (var item in ViewBag.students)

{

<div>ID: @item.Id </div>

<div>Name: @item.Name </div>

<div>Fee: @item.Fee </div>

}

<hr />

<table class="table">

<tr>

<th>

Exam SL

</th>

<th>

Exam Name

</th>

<th>

Board

</th>

<th>

Institution

</th>

<th>

Result

</th>

<th></th>

</tr>

@foreach (var item in ViewBag.results)

{

<tr>

<td>

@item.Examsl

</td>

<td>

@item.Examname

</td>

<td>

@item.Board

</td>

<td>

@item.Institution

</td>

<td>

@item.Result

</td>

</tr>

}

</table>

<hr />

<table class="table">

<tr>

<th>

Voucher No

</th>

<th>

Name

</th>

<th>

Date

</th>

<th>

Head

</th>

<th>

Amount

</th>

</tr>

@foreach (var item in ViewBag.fees)

{

<tr>

<td>

@item.Voucherno

</td>

<td>

@item.Student.Name

</td>

<td>

@item.InputDate

</td>

<td>

@item.Headname

</td>

<td>

@item.Amount

</td>

</tr>

}

</table>

<div>

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

1. Run the project and enjoy.
2. Controller>Right Click>Add>Controller> API controller with actions, using entity framework>add
3. Model: Items (…) and DataContext Class: existing…>Controller Name: itemsApiController>Add
4. Test the api using <http://localhost:49803/api/itemsApi> and postman.
5. Controller>Right Click>Add Controller> ItemsCrudController(Empty)>Add
6. Right Click Index in ItemsCrudController.cs>Add view>Empty>Add
7. Fetch all environment tags(css+js files) into <head> in \_Layout.cshtml
8. Write Codes in Views/ItemsCrud/index.cshtml as:

@{

ViewData["Title"] = "Index";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<script type="text/javascript">

$(document).ready(function () {

$("body").on('click', '#ed1', function () {

//alert("OK");

$("#t1").val($(this).closest('tr').children(":nth-child(1)").html());

$("#t2").val($(this).closest('tr').children(":nth-child(2)").html());

$("#t3").val($(this).closest('tr').children(":nth-child(3)").html());

})

function blank() {

$("#t1").val("");

$("#t2").val("");

$("#t3").val("");

}

function refresh() {

$.ajax({

type: "get",

url: "http://localhost:49803/api/itemsapi",

datatype: "json",

contenttype: "application/json",

success: function (data) {

var s = "";

for (var i = 0; i < data.length; i++) {

s += "<tr>";

s += "<td>" + data[i].itemcode + "</td>";

s += "<td>" + data[i].itemname + "</td>";

s += "<td>" + data[i].picture + "</td>";

s += "<td>" + "<a href='#' class='btn btn-primary' id='ed1'>Edit</a>" + "</td>";

s += "</tr>"

}

$("#student tbody").html(s);

},

error: function (xhr) {

alert(xhr.responsetext);

}

});

}

refresh();

$("#b1").click(function () {

var productData = {

"itemcode": $("#t1").val(),

"itemname": $("#t2").val(),

"picture": $("#t3").val()

};

$.ajax({

type: "POST",

url: "http://localhost:49803/api/itemsapi",

data: JSON.stringify(productData),

dataType: "json",

contentType: "application/json",

success: function (data) {

alert("Data Inserted Successfully");

blank();

refresh();

},

error: function (xhr) {

alert(xhr.responseText);

}

});

});

$("#b3").click(function () {

var productData = {

"itemcode": $("#t1").val(),

"itemname": $("#t2").val(),

"picture": $("#t3").val()

};

$.ajax({

type: "PUT",

url: "http://localhost:49803/api/itemsapi/" + $("#t1").val(),

data: JSON.stringify(productData),

dataType: "json",

contentType: "application/json",

success: function (data) {

alert("Data Inserted Successfully");

blank();

refresh();

},

error: function (xhr) {

alert(xhr.responseText);

}

});

});

$("#b2").click(function () {

var productData = {

"itemcode": $("#t1").val(),

"itemname": $("#t2").val(),

"picture": $("#t3").val()

};

$.ajax({

type: "DELETE",

url: "http://localhost:49803/api/itemsapi/" + $("#t1").val(),

data: JSON.stringify(productData),

dataType: "json",

contentType: "application/json",

success: function (data) {

alert("Data Inserted Successfully");

blank();

refresh();

},

error: function (xhr) {

alert(xhr.responseText);

}

});

});

});

</script>

<div class="container">

<h2>CRUD On ITEMS(WITHOUT UPLOAD)</h2>

<div class="form-group">

<label for="t1" class="col-md-2">Item code:</label>

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

<input type="text" id="t1" class="form-control" />

</div>

</div>

<div class="form-group">

<label for="t2" class="col-md-2">Item Name:</label>

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

<input type="text" id="t2" class="form-control" />

</div>

</div>

<div class="form-group">

<label for="t3" class="col-md-2">Picture:</label>

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

<input type="text" id="t3" class="form-control" />

</div>

</div>

<div class="form-group">

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

<input type="button" id="b1" value="Add" class="btn btn-default" />

<input type="button" id="b2" value="Delete" class="btn btn-default" />

<input type="button" id="b3" value="Update" class="btn btn-default" />

</div>

</div>

</div>

<hr />

<table class="table table-striped" id="student">

<thead>

<tr>

<th>Item Code</th>

<th>Item Name</th>

<th>Picture</th>

<th>Action</th>

</tr>

</thead>

<tbody></tbody>

</table>

1. Add line in \_Layout.cshtml as:

<li><**a** **asp-area**="" **asp-controller**="ItemsCrud" **asp-action**="Index">Items Crud</**a**></li>

1. Run the project and enjoy………………..