ASP .NET CORE WEB APP (model -view – controller)

Може да има например за [Range(100,110)] или [StringLength(20)].

Създава се допълнителен клас който се казва нали както е зададено в задачата за типове на изброяване например TypeRoom, там променяме класа и го кръщаваме горе public enum TypeRoom, и изброяваме това което ни е зададено. Дефинира в нормалния клас public TypeRoom TypeRoom {get; set;}

Пример за един нормален модел: Името е според задачата на класа както и public enum-a после.

public class Room

{

public int Id { get; set; }

[Range(100,110)]

public int NumberOfRoom { get; set; }

public decimal Price { get; set; }

public string Image { get; set; }

public TypeRoom TypeRoom { get; set; }

public int NumberOfNights {get; set;}

public DateTime Start { get; set; }

public DateTime End { get; set; }

}

}

public enum TypeRoom

{

Edinichna, dvoichna , studio

}

2. Създаваме контролер - търсим папката Controllers, десен бутон Add Контролер, избираме MVC Conroller with views, using Entity Framework, след това си избираме класа и дата контекста го избираме.

3. Създаване на миграция Add-Migration InitialCreate, Update-DataBase , добавя се от Tools -> Nuget Package Manager -> Package Manager Console Ако не тръгва изтриваме БД и нова миграция. Трием я като влезем в Sql server object explorer намирам я и трия. Ако трябва да сменя името и, public DbSet<Hotelska\_staq.Models.Room> Room { get; set; }

4. Layout - Трябват два индекса единия за клиента другия за админа в класа както се казва Модела и така свързвам Layout, след това трябва да отидем в Controllers HomeController и да копираме public IActionResult Index , и да сложим абсолютно същия само че с името ни на Indexa за потребителите:

public IActionResult IndexKlient()

{

return View();

}

После отиваме в RoomsController тоест както ни се казва класа и: по същия начин

@if (!this.User.Identity.IsAuthenticated)

{

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

}

else

if (this.User.IsInRole("Admin"))

{

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Rooms" asp-action="Index">DanniZaStaq</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Rooms" asp-action="IndexKlient">StaiZaKlient</a>

</li>

}

else

{

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" asp-area="" asp-controller="Rooms" asp-action="IndexKlient">StaiZaKlient</a>

</li>

}

</ul>

5. Identity - Отиваме Areas-> Add -> New Scaffolded item -> Identity -> Add -> след това избираме Log in, Log out, Register, отдолу DbContext след това оправяме Register.cshtml.cs,register.cshtml.,login.cshtml,login.cshtml.cs. Първо започваме с register.cshtml.cs в него под

[Required]

[EmailAddress]

[Display(Name = "Email")]

public string Email { get; set; }

поставяме public string UserName { get; set; } След това слизаме надолу и търсим Input UserName = UserName си го поправяме и малко по надолу след това поставяме :

if (result.Succeeded)

{

\_logger.LogInformation("User created a new account with password.");

if (\_userManager.Users.Count() == 1)

{ await \_userManager.AddToRoleAsync(user, "Admin"); }

else

{ await \_userManager.AddToRoleAsync(user, "User"); }

След това отиваме вече в register.cshtml там единствени над емайла по аналагоия на него си добавяме и за UserName:

<div class="form-group">

<label asp-for="Input.UserName"></label>

<input asp-for="Input.UserName" class="form-control" placeholder =" UserName">

<span asp-validation-for="Input.UserName" class="text-danger"></span>

</div>

На всеки един ред добавяме placeholder за красота placeholder =" UserName">

както тука така и в долните редове се добавя! за емейл, парола, потвърждаване на парола.

Преминаваме към login.cshtml.cs там обаче това за емайла който беше за register.cshtml.cs:

[Required]

[EmailAddress]

[Display(Name = "Email")]

public string Email { get; set; } го поставяме в коментар и добавяме под него само public string UserName { get; set; } след това пак слизаме надолу вместо Input.Email го заместваме с UserName.

След това отиваме в login.cshtm и там вместо емайл просто го променяме на UserName всичко друго си остава също така си добавяме и placeholder както при register.cshtml.

6. Сървиси те се слагат в Startup.cs:

public Startup(IConfiguration configuration)

{

Configuration = configuration;

}

public IConfiguration Configuration { get; }

// This method gets called by the runtime. Use this method to add services to the container.

public void ConfigureServices(IServiceCollection services)

{

services.AddDbContext<ApplicationDbContext>(options =>

options.UseSqlServer(

Configuration.GetConnectionString("DefaultConnection")));

//services.AddDefaultIdentity<IdentityUser>(options => options.SignIn.RequireConfirmedAccount = true)

// .AddEntityFrameworkStores<ApplicationDbContext>();

{

services.AddIdentity<IdentityUser, IdentityRole>().AddEntityFrameworkStores<ApplicationDbContext>().AddDefaultTokenProviders().AddDefaultUI();

services.Configure<IdentityOptions>(options =>

{

options.Password.RequireDigit = false;

options.Password.RequireLowercase = false;

options.Password.RequireNonAlphanumeric = false;

options.Password.RequireUppercase = false;

options.Password.RequiredLength =

options.Password.RequiredUniqueChars = 0;

options.User.RequireUniqueEmail = true;

});

}

services.AddControllersWithViews();

services.AddRazorPages();

}

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

using (var serviceScope = app.ApplicationServices.CreateScope())

{

using (var context =

serviceScope.ServiceProvider.GetRequiredService<ApplicationDbContext>())

{

context.Database.Migrate();

if (!context.Roles.Any())

{

context.Roles.Add(new IdentityRole

{

Name = "Admin",

NormalizedName = "ADMIN",

});

context.Roles.Add(new IdentityRole

{

Name = "User",

NormalizedName = "USER",

});

context.SaveChanges();

}

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

app.UseDatabaseErrorPage();

}

else

{

app.UseExceptionHandler("/Home/Error");

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseAuthentication();

app.UseAuthorization();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

endpoints.MapRazorPages();

});

}

}

}

}

}

7. Авторизация в Home Index:

<style>

img {

width: 50%;

border: 2px solid blue;

}

.jumbotron{background-color:lightgreen}

#sexy {

width: 50%;

border: 2px solid pink;

margin-left: 256px;

}

#sexyy {

width: 50%;

border: 2px solid blue;

margin-left: 200px;

}

.display-4

{

text-align:center;

color: pink;

}

h2

{

color:lightpink;

text-align:center;

}

</style>

@if (!this.User.Identity.IsAuthenticated)

{

<div class="text-center">

<h1 class="display-4">Добре дошли! Моля идентифицирайте се!</h1>

<img src="~/Images/sexy.jpg" />

</div>

}

else

{

if (this.User.IsInRole("Admin"))

{<div><h2>Have a nice day, Admin!</h2></div>

<img id="sexy" src="~/Images/ulala.jpg" />

}

else

{

<div class="jumbotron bg-eventures w-75 mx-auto">

<h1 class="text-center text-secondary">Welcome, @this.User.Identity.Name</h1>

<h1 class="display-4">Тук се пише текст специално за задачата: </h1>

<p>

<h4>Тук се пише текст специално за задачата: </h4>

Тук се пише текст специално за задачата:

</p>

<img id ="sexyy"src="~/Images/sexy.jpg" />

</div>

}

}

//Тук се поставя снимката спрямо задачата като направо може да я вземеш от wwwroot и ти се поставя както е долу:

Обърни внимание на img id = " " src =

Този ид се използва за да му се зададе име и да се стилизира после не ти е нужен! Но може да го направиш да изглежда добре.

Така сега започва оправянето на всичко!

Privacy добави някъв текст да има нещо.

Create единствено добавяш опциите си от класа за енума: Търсиш класа както се казва и на втория ред поставяш опциите:

<select asp-for="TypeRoom" class="form-control">

<option>Edinichna</option>

<option>Dvoina</option>

<option>Studio</option>

</select>

Delete не се бута нищо

При Details там трябва да имаш два един за админа другия за клиента кръщаваш втория DetailsKlient:

В DetailsKlient си добавяш

<style>

img {

width: 50%;

}

</style>

После слизаш надолу и буташ втория ModelImage:

@\*@Html.DisplayFor(model => model.Image)\*@ слагаш само него в коментар единия ред и отдолу пишеш

<img src="@Model.Image" />

Слизаш най надолу и махаш онези бутончета Edit Delete и т.н оставяш само

<a asp-action="IndexKlient">Back to List</a> обърни внимание на името на индекса трябва да ти е на клиента.

<div>

<h4>Room</h4>

<hr />

<dl class="row">

<dt class="col-sm-2">

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

</dt>

<dd class="col-sm-10">

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

</dd>

<dt class="col-sm-2">

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

</dt>

<dd class="col-sm-10">

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

</dd>

<dt class="col-sm-2">

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

</dt>

<dd class="col-sm-10">

@\*@Html.DisplayFor(model => model.Image)\*@

<img src="@Model.Image" />

</dd>

<dt class="col-sm-2">

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

</dt>

<dd class="col-sm-10">

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

</dd>

<dt class="col-sm-2">

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

</dt>

<dd class="col-sm-10">

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

</dd>

<dt class="col-sm-2">

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

</dt>

<dd class="col-sm-10">

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

</dd>

<dt class="col-sm-2">

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

</dt>

<dd class="col-sm-10">

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

</dd>

</dl>

</div>

<div>

@\*<a asp-action="Edit" asp-route-id="@Model.Id">Edit</a> |\*@

<a asp-action="IndexKlient">Back to List</a>

</div>

След това отиваш в другия истинския Details за админа. И там по същия начин всичко както в клиента само, че долу е

<a asp-action="Edit" asp-route-id="@Model.Id">Edit</a> |

<a asp-action="Index">Back to List</a>

В Edit там единствено добавяш опциите си от класа за енума: Търсиш класа както се казва и на втория ред поставяш опциите:

<select asp-for="TypeRoom" class="form-control">

<option>Edinichna</option>

<option>Dvoina</option>

<option>Studio</option>

</select>

Преминаваме към Index който си е главния за Админа и за всичко при него: при първия <thead> не пипаме нищо освен единствено най долу където добавяме: <th>Трайност до края на промоцията</th>

След това минаваме на долните пак отново след foreach-a и отиваме на Image и там реда го отбелязваме @\*@Html.DisplayFor(modelItem => item.Image)\*@ и го слагаме в коментар: и отдолу пишем: <img src="@item.Image" />

След това най - долу създаваме ново <td>:

<td>

@(item.End.Day - item.Start.Day ) days

</td>

И най най отдолу ако искам да имам бутончета поставям това :

<td>

<button id="buton"> <a asp-action="Edit" asp-route-id="@item.Id">Edit</a> </button>

</td>

<td><button id="buton"> <a asp-action="Details" asp-route-id="@item.Id">Details</a></button></td>

<td><button id="buton"> <a asp-action="Delete" asp-route-id="@item.Id">Delete</a></button></td>

}

</tbody>

</table>

Преминаваме към IndexKlient: В него в началото пак в <thead> добавяме <th>Трайност до края на промоцията</th> и слагаме в коментар za datata start и end

В foreach по същия начин : @\*@Html.DisplayFor(modelItem => item.Image)\*@ и го слагаме в коментар: и отдолу пишем: <img src="@item.Image" /> , там пак датата и края на датата се слагат в коментар, и се поставя същото <td> от предишния индекс:

<td>

@(item.End.Day - item.Start.Day ) days

</td>

Датите се слагат в коментар!

Най - отдолу

<td>

@\*<a asp-action="Edit" asp-route-id="@item.Id">Edit</a> |\*@

<a asp-action="DetailKlient" asp-route-id="@item.Id">Details</a> |

@\*<a asp-action="Delete" asp-route-id="@item.Id">Delete</a>\*@

</td>

Слагат се в коментари и Details трябва да се преименува на DetailsKlient както е направено.

Относно стилизацията: Примерна за Layout:

<style>

nav {

background-image: url(/Images/sexy.jpg);

}

body {

background-image: url(/Images/sexy.jpg);

}

</style>

!. това е ако не успея и не ми се занимава:

<style>

.container {

background-color: mistyrose;

color: aquamarine;

background-image:url(/Images/sexy.jpg);

background-size: cover;

}

div {

background-color: cornflowerblue;

}

footer {

background-color: darkgrey;

color: red;

}

nav {

background-color: lightpink;

}

</style>