

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

12.03.2023

MVC CORE 7.0

VISUAL STUDIO 2022

Several thin, curved, light blue lines that sweep upwards from the bottom left corner of the page.

Cengiz GÜNGÖRDÜ

TUZLA MESLEKİ VE TEKNİK ANADOLU LİSESİ

İçindekiler

Table of Contents

| | |
|-------------------------------------------------------------------------------------------------------------|----|
| 1-core webpage..... | 2 |
| 2- nuget..... | 3 |
| 3-appsettings.json a connection string eklenir..... | 3 |
| 4_person class ı..... | 3 |
| 5_ context clasının içi..... | 3 |
| 6- program.cs de..... | 4 |
| 7- migration için nuget consolda aşağıdaki satır eklenir..... | 4 |
| 8-CREATE..... | 4 |
| 9-LİSTELEME..... | 6 |
| Home controller metot..... | 6 |
| VIEW Dosyasının içi..... | 7 |
| 10-SİLME..... | 7 |
| Home controller içi..... | 7 |
| 11-GÜNCELLEME..... | 8 |
| 12-ARAMA..... | 9 |
| 13- C.R.UD. A RESİM EKLE..... | 14 |
| Kayıt girişi View dosyası..... | 15 |
| Kayıt girişi kontroller dosyası..... | 15 |
| Güncelleme view dosyası..... | 16 |
| Güncelleme kontroller dosyası..... | 17 |
| 14 listeden Güncelleme view dosyası..... | 19 |
| View dosyası..... | 19 |
| Controller dosyası..... | 20 |
| 15- normal listeden sil link için ve listeden güncelle sil link için yazılmış kontrollerdeki actionlar..... | 22 |
| 16-navbardan arama..... | 23 |
| View dosyası layoutun içi..... | 23 |
| action dosyası içi..... | 23 |
| View dosyası home / index e eklenen satırlar..... | 23 |
| 17 Home index e Carousel ekleme..... | 23 |
| Home controllerdan sql bağlanmak için..... | 23 |

| | |
|--------------------------------------------------------|----|
| Bootstrap den Carousel şablonu alınır..... | 24 |
| Home index viewde liste türünde model oluşturulur..... | 24 |
| 18 Card görünümlü listeleme..... | 25 |
| CardListeleme controller..... | 25 |
| //bootstrapden alınan şablon..... | 26 |
| Cardlisteleme view in içi..... | 26 |
| 19 Somee.com dan free siteye yükleme..... | 26 |
| 20-KAYNAKÇA..... | 28 |

1-core webpage

Layout bootstrapden son versiyonu ile güncelle

Wwwroor klasörü altındakileri sil

<https://bootswatch.com/> dan aldığın bootstrapmin.cs yi kulan...

Navbar l da <https://bootswatch.com/> dan alarak shared klasörü içine

Add new item dan Razor View-Empty sayfaya yapıştırarak _layout içinden

```
<partial name="_nav" />
```

tag l ile ekleyelim.

2- nuget

Entityframeworkcore.tools

entityEntityframeworkcore.sqlserver

3-appsettings.json a connection string eklenir

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": { "conn": "data source=.;initial catalog=MvcCoreTutorial;integrated
security=true;TrustServerCertificate=True" }
//somee.com için
//"ConnectionStrings": {
  // "conn": "workstation id=dbtuzla.mssql.somee.com;packet size=4096;user
id=cengun_SQLLogin_1;pwd=8j6gomidj6;data source=dbtuzla.mssql.somee.com;persist security
info=False;initial catalog=dbtuzla;Connection
Timeout=520;TrustServerCertificate=True;Encrypt=false;MultiSubnetFailover=True"
  //},
}
```

4_person class ı

using System.ComponentModel.DataAnnotations;

namespace MvcCoreTutorial.Models.Domain

```
{
    public class Person
    {
        public int Id { get; set; }
        [Required]
        public string? Name { get; set; }
        [Required]
        [EmailAddress]
        public string? Email { get; set; }
    }
}
```

5_context clasının içi

using Microsoft.EntityFrameworkCore;

using System.Configuration;

namespace MvcCoreTutorial.Models.Domain

```
{
    public class DatabaseContext : DbContext
```

```

{
    public DatabaseContext(DbContextOptions<DatabaseContext> opts) :base(opts)
    {
    }
    public DbSet<Person> Person { get; set; }
}
}

```

6- program.cs de

`var app = builder.Build();` satırının üstüne aşağıdaki servis eklenir.

```

builder.Services.AddDbContext<DatabaseContext>(options =>
    options.UseSqlServer(builder.Configuration.GetConnectionString("conn")));

```

7- migration için nuget consolda aşağıdaki satır eklenir.

Add-migration init

8-sqlde database için nuget consolda aşağıdaki satır eklenir.

update-database

8-CREATE

Home controller altında ilk hali HttpPost kısmı geliştirilecek az sonra..

```

    public IActionResult AddPerson()
    {
        return View();
    }

    [HttpPost]
    public IActionResult AddPerson(Person person)
    {
        return View();
    }

```

“Addperson.cshtml” dosyasının içi

@model tuzlaCore_tut.Models.Person

<div class="mt-2" style="width:50%">

```

<form asp-action="AddPerson">
  <h2>Add Person</h2>
  <div class="mb-3">
    <label asp-for="Name" class="form-label">Name</label>
    <input asp-for="Name" type="text" class="form-control">
    <span class="text-danger" asp-validation-for="Name"></span>
  </div>

  <div class="mb-3">
    <label asp-for="Email" class="form-label">Email</label>
    <input asp-for="Email" type="text" class="form-control">
    <span class="text-danger" asp-validation-for="Email"></span>
  </div>

  @if (TempData["msg"] != null)
  {
    <div class="mb-3">
      @TempData["msg"]
    </div>
  }
  <button type="submit" class="btn btn-primary">Submit</button>
  <a href="/home/DisplayPersons" class="btn btn-primary">All Persons</a>
</form>
</div>

```

DB bağlantısı için homecontroller içine dependency injection ile bağlantı kuruyoruz.

Yapıcı konstraktır yardımı ile //ctor

```

private readonly DatabaseContext _ctx;
public HomeController(DatabaseContext ctx)
{
  _ctx = ctx;
}

```

Model state control edilerek gelen veriler control ediliyor. Hatalı ise validation yardımı ile view page e geri gönderiliyor ve db ye kayıt işlemi yapılıyor.

```
//it is get method
public IActionResult AddPerson()
{
    return View();
}

[HttpPost]
public IActionResult AddPerson(Person person)
{
    if(!ModelState.IsValid)
    {
        return View();
    }
    try
    {
        _ctx.Person.Add(person);
        _ctx.SaveChanges();
        TempData["msg"] = "Added successfully";
        return RedirectToAction("AddPerson");
    }
    catch (Exception ex)
    {
        TempData["msg"] = "Could not added!!!";
        return View();
    }
}
}
```

9-LISTELEME

Home controller metot

```
public IActionResult DisplayPersons()
{
    var persons = _ctx.Person.ToList();
    return View(persons);
}
```

VIEW Dosyasının içi

@model IEnumerable<tuzlaCore_tut.Models.Person>

```
<div class="mt-2" style="width:50%">
  <div class="row">
    <h2>Person</h2>
    <div class="col-md-4 m-3">
      <a href="/Home/AddPerson" class="btn btn-primary">Add Person</a>
    </div>
    <table class="table table-striped">
      <thead>
        <tr>
          <th>Name</th>
          <th>Email</th>
          <th>Action</th>
        </tr>
      </thead>
      <tbody>
        @foreach (var person in Model)
        {
          <tr>
            <td>
              @person.Name
            </td>
            <td>@person.Email</td>
            <td>
              <a class="btn btn-success" href="/Home/EditPerson?id=@person.Id">Edit</a>
              <a class="btn btn-danger" href="/Home/DeletePerson?id=@person.Id">Delete</a>
            </td>
          </tr>
        }
      </tbody>
    </table>
  </div>
</div>
```

10-SİLME

Home controller içi

```
public IActionResult DeletePerson(int id)
{
    try
    {
        var person = _ctx.Person.Find(id);
        if (person != null)
        {
            _ctx.Person.Remove(person);
        }
    }
}
```



```

        _ctx.SaveChanges();
    }
}
catch (Exception ex)
{

}

return RedirectToAction("DisplayPersons");
}

```

11-GÜNCELLEME

@model tuzlaCore_tut.Models.Person

```
@{
}
```

```

<div class="mt-2" style="width:50%">
    <form asp-action="EditPerson">
        <input asp-for="Id" type="hidden" class="form-control">

        <h2>Edit Person</h2>
        <div class="mb-3">
            <label asp-for="Name" class="form-label">Name</label>
            <input asp-for="Name" type="text" class="form-control">
            <span class="text-danger" asp-validation-for="Name"></span>
        </div>

        <div class="mb-3">
            <label asp-for="Email" class="form-label">Email</label>
            <input asp-for="Email" type="text" class="form-control">
            <span class="text-danger" asp-validation-for="Email"></span>
        </div>

        @if (TempData["msg"] != null)
        {
            <div class="mb-3">
                @TempData["msg"]
            </div>
        }
        <button type="submit" class="btn btn-primary">Update</button>
        <a href="/Home/DisplayPersons" class="btn btn-primary">Back</a>
    </form>
</div>

```

EditPerson

Home konroller altındaki metot

```

public IActionResult EditPerson(int id)
{
    var person = _ctx.Person.Find(id);
    return View(person);
}

```

```

[HttpPost]
public IActionResult EditPerson(Person person)
{
    if (!ModelState.IsValid)
    {
        return View();
    }
    try
    {
        _ctx.Person.Update(person);
        _ctx.SaveChanges();
        return RedirectToAction("DisplayPersons");
    }
    catch (Exception ex)
    {
        TempData["msg"] = "Could not update!!!";
        return View();
    }
}

```

12-ARAMA

Girilen isme göre arama

```

public IActionResult FindPerson()
{
    return View();
}

```

```

[HttpPost]
public IActionResult FindPerson(string Name)
{
    if (Name!=null)
    {
        var foundpersons = (from i in _ctx.Person
                           where i.Name.StartsWith(Name)
                           select i).ToList();
        if (foundpersons.Count>0)
        {
            return View("DisplayPersons", foundpersons);
        }
        TempData["msg"] = "Could not found!!!";
        return View();
    }
    else
    {
        TempData["msg"] = "Please input a Name!!!";
        return View();
    }
}

```

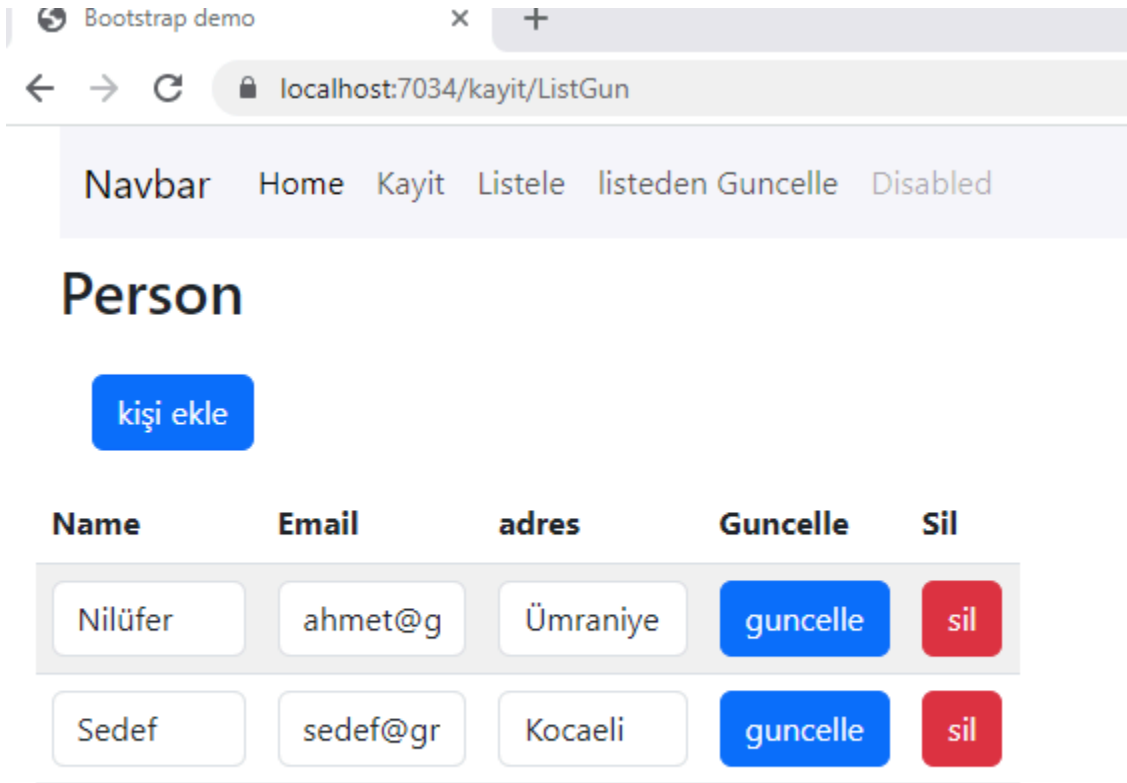
View sayfası

```

<div class="mt-2" style="width:50%">
  <form asp-action="FindPerson">
    <h2>Find Person</h2>
    <div class="mb-3">
      <label name="Name" class="form-label">Name</label>
      <input name="Name" type="text" class="form-control">
    </div>
    @if (TempData["msg"] != null)
    {
      <div class="mb-3">
        @TempData["msg"]
      </div>
    }
    <button type="submit" class="btn btn-primary">Find </button>
  </form>
</div>

```

LİSTELEME İÇİNDE GÜNCELLEME



VIEW DOSYASI

@model List<core70repeat.Models.personel>

```
<div class="mt-2" style="width:50%">
  <div class="row">
    <h2>Person</h2>
    <div class="col-md-4 m-3">
      <a href="/kayit/kisi_ekle" class="btn btn-primary">kişi ekle</a>
    </div>
```

```
@if (TempData["msg"] != null)
```

```
{
```

```
    <div class="mb-3">
```

```
        @TempData["msg"]
```

```
    </div>
```

```
}
```

```
<table class="table table-striped">
```

```
    <thead>
```

```
        <tr>
```

```
            <th>Name</th>
```

```
            <th>Email</th>
```

```
            <th>adres</th>
```

```
            <th>Guncelle</th>
```

```
            <th>Sil</th>
```

```
        </tr>
```

```
    </thead>
```

```
    <tbody>
```

```
        @foreach (var person in Model)
```

```
        {
```

```
            <form asp-controller="kayit" asp-action="ListGun" method="post">
```

```
                <input name="Id" value="@person.Id" type="hidden" class="form-control">
```

```
            <tr>
```

```
                <td>
```

```
                    <input name="Name" value="@person.Name" type="text" class="form-control">
```

```

        </td>

        <td>

            <input name="Email" value="@person.Email" type="text" class="form-control">

        </td>

        <td>

            <input name="Adres" value="@person.Adres" type="text" class="form-control">

        </td>

        <td>

            <button type="submit" class="btn btn-primary">guncelle</button>

        </td>

        <td>

            <a class="btn btn-danger" href="/kayit/sil?id=@person.Id">sil</a>

        </td>
    </tr>

</form>

    }
</tbody>
</table>
</div>

</div>

```

KONTROLLER

```
public IActionResult ListGun()
```

```
{
```

```
    var listelenecekler = _ctx.personeller.ToList();
```

```
    return View(listelenecekler);
```

```
}
```

```
[HttpPost]
```

```
public IActionResult ListGun(personel yeni_perseonel)
```

```
{
```

```
    if (!ModelState.IsValid)
```

```
    {
```

```
        TempData["msg"] = "Lütfen eksik bilgi girmeyin!!!";
```

```
        var listelenecekler = _ctx.personeller.ToList();
```

```
        return View(listelenecekler);
```

```
    }
```

```
    try
```

```
    {
```

```
        _ctx.personeller.Update(yeni_perseonel);
```

```
        _ctx.SaveChanges();
```

```
        return RedirectToAction("ListGun");
```

```
    }
```

```
    catch (Exception)
```

```
    {
```

```
        TempData["msg"] = "Bilinmeyen hata!!! Tekrar deneyin";
```

```
        var listelenecekler = _ctx.personeller.ToList();
```

```
        return View(listelenecekler);
```

```
    }
```

```
}
```

13- C.R.UD. A RESİM EKLE

Kayıt girişi View dosyası

@model core70repeat.Models.personel

```
<div class="mt-2" style="width:50%">
  <form asp-action="kisi_ekle" method="post" enctype="multipart/form-data">
    <h2>Kişi ekle</h2>
    <div class="mb-3">
      <label asp-for="Name" class="form-label">Name</label>
      <input asp-for="Name" type="text" class="form-control">
      <span class="text-danger" asp-validation-for="Name"></span>
    </div>

    <div class="mb-3">
      <label asp-for="Adres" class="form-label">Adres</label>
      <input asp-for="Adres" type="text" class="form-control">
      <span class="text-danger" asp-validation-for="Adres"></span>
    </div>

    <div class="mb-3">
      <label asp-for="Resim" class="form-label">Resim</label>
      <input name="formfile" type="file" class="form-control">
      <span class="text-danger" asp-validation-for="Resim"></span>
    </div>

    @if (TempData["msg"] != null)
    {
      <div class="mb-3 text-danger">
        @TempData["msg"]
      </div>
    }

    <div class="mb-3">
      <label asp-for="Email" class="form-label">Email</label>
      <input asp-for="Email" type="text" class="form-control">
      <span class="text-danger" asp-validation-for="Email"></span>
    </div>

    <button type="submit" class="btn btn-primary">Submit</button>
    <a href="/kayit/listele" class="btn btn-primary">Tümünü Göster</a>
  </form>
</div>
```

Kayıt girişi kontroller dosyası

```
public IActionResult kisi_ekle()
{
  return View();
}
```



```

string? randomName;
[HttpPost]
public IActionResult kisi_ekle(personel yeni_perseonel, IFormFile formfile)
{
    ModelState.Remove("formfile");

    if (!ModelState.IsValid)
    {
        return View(yeni_perseonel);
    }

    if (formfile != null)
    {
        var extent = Path.GetExtension(formfile.FileName);
        randomName = $"{Guid.NewGuid()}{extent}";
        var path = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot\\img", randomName);

        using (var stream = new FileStream(path, FileMode.Create))
        {
            formfile.CopyTo(stream);
        }
    }
    else
    {
        TempData["msg"] = "resim girin";
        return View(yeni_perseonel);
    }
}

```

```

try
{
    yeni_perseonel.Resim = randomName; // rasgele üretilmiş resim adı kayıt edilir
    _ctx.personeller.Add(yeni_perseonel);
    _ctx.SaveChanges();
    TempData["msg"] = "kayıt başarılı";
    return RedirectToAction("kisi_ekle");
}
catch (Exception)
{
    TempData["msg"] = "kayıt başarılı değil";
    return View(yeni_perseonel);
}
}

```

Güncelleme view dosyası

@model core70repeat.Models.personel

<div class="mt-2" style="width:50%">

```

<div class="btn-primary">
  <h1>GÜNCELLEME SAYFASI</h1>
</div>

<form asp-action="guncelle" method="post" enctype="multipart/form-data">
  <h2>Add Person</h2>

  <input asp-for="Id" type="hidden" class="form-control" >
  <input asp-for="Resim" type="hidden" class="form-control">

  <div class="mb-3">
    <label asp-for="Name" class="form-label">Name</label>
    <input asp-for="Name" type="text" class="form-control">
    <span class="text-danger" asp-validation-for="Name"></span>
  </div>

  <div class="mb-3">
    <label asp-for="Adres" class="form-label">Adres</label>
    <input asp-for="Adres" type="text" class="form-control">
    <span class="text-danger" asp-validation-for="Adres"></span>
  </div>

  <div class="mb-3">
    

    <input name="formfile" type="file" class="form-control">
    @if (TempData["msg"] != null)
    {
      <div class="mb-3 text-danger">
        @TempData["msg"]
      </div>
    }
    <span class="text-danger" asp-validation-for="Resim"></span>
  </div>

  <div class="mb-3">
    <label asp-for="Email" class="form-label">Email</label>
    <input asp-for="Email" type="text" class="form-control">
    <span class="text-danger" asp-validation-for="Email"></span>
  </div>

  <button type="submit" class="btn btn-primary">güncelle</button>
  <a href="/kayit/listele" class="btn btn-primary">Tümünü Göster</a>
</form>
</div>

```

Güncelleme kontroller dosyası

```

public IActionResult guncelle(int id)
{

```

```

        var guncellenecek_kayit = _ctx.personeller.Find(id);
        return View(guncellenecek_kayit);
    }

    [HttpPost]
    public IActionResult guncelle(personel yeni_perseoneel, IFormFile formfile)
    {
        ModelState.Remove("formfile");

        if (!ModelState.IsValid)
        {
            return View(yeni_perseoneel);
        }

        if (formfile != null)
        {
            //eski resim dosyasını sil
            string eski_resim = yeni_perseoneel.Resim;
            //www/img klasöründen sil
            var path_sil = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot\\img", eski_resim);

            if (System.IO.File.Exists(path_sil))
            {
                System.IO.File.Delete(path_sil);
            }

            //yeniye ekle
            var extent = Path.GetExtension(formfile.FileName);
            randomName = $"{Guid.NewGuid()}{extent}";
            var path = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot\\img", randomName);

            using (var stream = new FileStream(path, FileMode.Create))
            {
                formfile.CopyTo(stream);
            }

            //yeni_perseoneel.Resim = string.IsNullOrEmpty(randomName) ? yeni_perseoneel.Resim :
randomName;
            yeni_perseoneel.Resim = randomName;
        }

        try
        {
            _ctx.personeller.Update(yeni_perseoneel);
            _ctx.SaveChanges();

            return RedirectToAction("listele");
        }
        catch (Exception)
        {
            TempData["msg"] = "güncelleme başarılı değil";
            return View(yeni_perseoneel);
        }
    }

```

```
}  
}
```

14 listeden Güncelleme view dosyası

View dosyası

```
@model List<core70repeat.Models.personel>
```

```
<div class="mt-2" style="width:90%">
```

```
<div class="row">
```

```
<div class="col-md-8 m-3" class="btn btn-primary">
```

```
<h2>Direk listeden tüm alanları güncelleyebilirsiniz</h2>
```

```
</div>
```

```
@if (TempData["msg"] != null)
```

```
{
```

```
<div class="mb-3">
```

```
@TempData["msg"]
```

```
</div>
```

```
}
```

```
<table class="table table-striped">
```

```
<thead>
```

```
<tr>
```

```
<th>Name</th>
```

```
<th>Email</th>
```

```
<th>adres</th>
```

```
<th>resim</th>
```

```
<th>Yeni resim</th>
```

```
<th>Guncelle</th>
```

```
<th>Sil</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
@foreach (var person in Model)
```

```
{
```

```
<form asp-controller="kayit" asp-action="ListGun" method="post"
```

```
enctype="multipart/form-data">
```

```
<input name="Id" value="@person.Id" type="hidden" class="form-control">
```

```
<input name="Resim" value="@person.Resim" type="hidden" class="form-control">
```

```
<tr>
```

```
<td>
```

```
<input name="Name" value="@person.Name" type="text" class="form-control">
```

```
</td>
```

```
<td>
```

```

        <input name="Email" value="@person.Email" type="text" class="form-control">
    </td>
    <td>
        <input name="Adres" value="@person.Adres" type="text" class="form-control">
    </td>

    <td>
        

    </td>
    <td>

        <input name="formfile" type="file" class="form-control">
    </td>
    <td>
        <button type="submit" class="btn btn-primary">guncelle</button>

    </td>
    <td>
        <a class="btn btn-danger" href="/kayit/sil_lstgnci?id=@person.Id">sil</a>
    </td>
</tr>

</form>

    }
</tbody>
</table>
</div>

</div>

```

Controller dosyası

```

public IActionResult ListGun()
{
    var listelenecekler = _ctx.personeller.ToList();
    return View(listelenecekler);
}

```

```

}

[HttpPost]
public IActionResult ListGun(personel yeni_perseonel, IFormFile formfile)
{
    ModelState.Remove("formfile");

    if (!ModelState.IsValid)
    {
        TempData["msg"] = "Lütfen eksik bilgi girmeyin!!!";
        var listelenecekler = _ctx.personeller.ToList();
        return View(listelenecekler);
    }

    if (formfile != null)
    {
        //eski resim dosyasını sil
        string eski_resim = yeni_perseonel.Resim;
        //www/img klasöründen sil
        var path_sil = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot\\img", eski_resim);

        if (System.IO.File.Exists(path_sil))
        {
            System.IO.File.Delete(path_sil);
        }

        //yeniye ekle
        var extent = Path.GetExtension(formfile.FileName);
        randomName = $"{Guid.NewGuid()}{extent}";
        var path = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot\\img", randomName);

        using (var stream = new FileStream(path, FileMode.Create))
        {
            formfile.CopyTo(stream);
        }
        //yeni_perseonel.Resim = string.IsNullOrEmpty(randomName) ? yeni_perseonel.Resim :
randomName;
        yeni_perseonel.Resim = randomName;
    }

    try
    {
        _ctx.personeller.Update(yeni_perseonel);
        _ctx.SaveChanges();
        return RedirectToAction("ListGun");
    }
    catch (Exception)
    {
        TempData["msg"] = "Bilinmeyen hata!!! Tekrar deneyin";
    }
}

```

```

        var listelenecekler = _ctx.personeller.ToList();
        return View(listelenecekler);
    }
}

```

15- normal listeden sil link için ve listeden güncelle sil link için yazılmış controllerdeki actionlar

```

public IActionResult sil_istgncl(int id)
{
    var silinecek_veri = _ctx.personeller.Find(id);

    string randomName = silinecek_veri.Resim;
    //www/imdende sil
    var path = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot\\img", randomName);

    if (System.IO.File.Exists(path))
    {
        System.IO.File.Delete(path);
    }

    _ctx.personeller.Remove(silinecek_veri);
    _ctx.SaveChanges();
    return RedirectToAction("listGun");
}

```

```

public IActionResult sil(int id)
{
    var silinecek_veri = _ctx.personeller.Find(id);

    string randomName = silinecek_veri.Resim;
    //www/imdende sil
    var path = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot\\img", randomName);

    if (System.IO.File.Exists(path))
    {
        System.IO.File.Delete(path);
    }

    _ctx.personeller.Remove(silinecek_veri);
    _ctx.SaveChanges();
    return RedirectToAction("listele");
}

```

16-navbardan arama

View dosyası layoutun içi

```
<form asp-controller="kayit" asp-action="araNav" method="post" enctype="multipart/form-data"
class="d-flex" role="search">

    <input name="isim" class="form-control me-2" type="search" placeholder="isime göre ara"
aria-label="Search">

    <button class="btn btn-outline-success" type="submit">Search</button>

</form>
```

action dosyası içi

```
[HttpPost]
public IActionResult araNav(string isim)
{
    if (!string.IsNullOrEmpty(isim))
    {
        var arananlar = (from i in _ctx.personeller
                        where i.Name.StartsWith(isim)
                        select i).ToList();
        return View("listele", arananlar);
    }
    else
    {
        TempData["msg"] = "Lütfen aramak için isim giriniz!!!";
        return RedirectToAction("index", "home");
    }
}
```

Eğer veri girilmedi ise hata mesajı home/ indexte gösterildi

View dosyası home / index e eklenen satırlar

```
@if (TempData["msg"] != null)
{
    <div class="mb-3 text-danger">
        @TempData["msg"]
    </div>
}
```

17 Home index e Carousel ekleme (mevcut databaseden dinamik olarak ilk üç resim alınacaktır)

Home controllerdan sql bağlanmak için

```
private readonly DatabaseContext _ctx;

public HomeController(DatabaseContext ctx)
```



```
{
    _ctx = ctx;
}
```

```
public IActionResult Index()
{
    var ilk_uc_resim= _ctx.personeller.Take(3).ToList();
    return View(ilk_uc_resim);
}
```

//databaseden ilk üç kayıtlı resim çekilir ve index view ine gönderilir.

Bootstrap den Carousel Şablonu alınır.

```
<div id="carouselExample" class="carousel slide">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
  <button class="carousel-control-prev" type="button" data-bs-target="#carouselExample"
data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
  </button>
  <button class="carousel-control-next" type="button" data-bs-target="#carouselExample"
data-bs-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
  </button>
</div>
```

Home index viewde liste türünde model oluşturulur

```
@model List<core70repeat.Models.personel>
```

@{

```

    bool ilk_mi = false;
}

<div id="carouselExample" class="carousel slide">
  <div class="carousel-inner">

    @foreach (var kayit in Model)
    {
      if (ilk_mi==false)
      {
        <div class="carousel-item active">
          

        </div>
        ilk_mi = true;
      }
      else
      {
        <div class="carousel-item">
          

        </div>
      }
    }

    </div>

    <button class="carousel-control-prev" type="button" data-bs-target="#carouselExample"
data-bs-slide="prev">
      <span class="carousel-control-prev-icon" aria-hidden="true"></span>
      <span class="visually-hidden">Previous</span>
    </button>
    <button class="carousel-control-next" type="button" data-bs-target="#carouselExample"
data-bs-slide="next">
      <span class="carousel-control-next-icon" aria-hidden="true"></span>
      <span class="visually-hidden">Next</span>
    </button>
  </div>

```

18 Card görünümlü listeleme

CardListeleme controller

```

public IActionResult listelecard()
{
    var listelenecekler = _ctx.personeller.ToList();
    return View(listelenecekler);
}

```

//bootstrapden alınan şablon

```

<div class="card" style="width: 18rem;">
  
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">Some quick example text to build on the card title and make up the bulk of the
card's content.</p>
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
</div>

```

Cardlisteleme view in içi

```

@model List<core70repeat.Models.personel>
<div class="container text-center">
  <div class="row">
    @foreach (var person in Model)
    {
      <div class="col-md-4 mt-5">
        <div class="card" style="width: 18rem;">
          
          <div class="card-body">
            <h5 class="card-title">@person.Name</h5>
            <p class="card-text">@person.Adres </p>
            <a class="btn btn-success" href="/home/index">Anasayfa</a>
          </div>
        </div>
      </div>
    }
  </div>
</div>

```

19 Somee.com dan free siteye yükleme

Database oluştur. Connection stringi oradan alıp c sahrpda **Appsetting.json** eklenir.

Bu alınmış con string cümlesinin devamına

Timeout=520;TrustServerCertificate=True;Encrypt=false;MultiSubnetFailover=True

Ekleyerek aşağıdaki hale getir

Appsetting.json içi

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "conn": "workstation id=dbtuzla.mssql.somee.com;packet size=4096;user
id=cengun_SQLLogin_1;pwd=8j6gomidj6;data source=dbtuzla.mssql.somee.com;persist security
info=False;initial catalog=dbtuzla;Connection
Timeout=520;TrustServerCertificate=True;Encrypt=false;MultiSubnetFailover=True"
  }
}

} //workst
```

Program.cs de yapılan değişiklik

```
builder.Services.AddDbContext<DatabaseContext>(options =>
{
    options.UseSqlServer(builder.Configuration.GetConnectionString("conn"), option =>
    {
        option.EnableRetryOnFailure();
    });
});
```

Localden sharp içinden

Add-migrations init

Ve update-database yazarak somee.com da database oluşumu gerçekleşir.

Sonra

Proje publish edilerek, çıkan dosyalar filezilla ile web sitesine yüklenir

20-KAYNAKÇA

<https://www.youtube.com/watch?v=e2I7EzuCt1g>