

Opgave: Uge 4 – Anton's Auto

Fag: Udviklingsmiljøer

Udarbejdet af: Lars Larsen

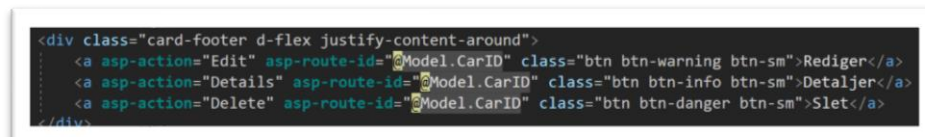
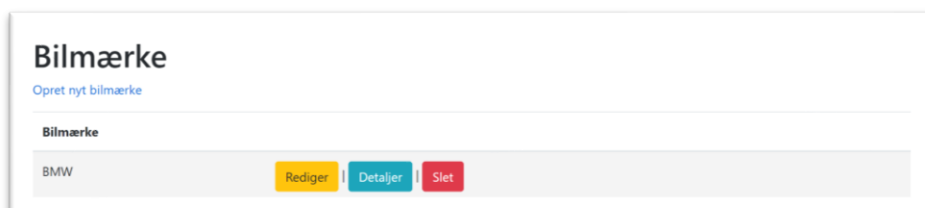
I denne uge har jeg arbejdet med

- Opdatering af HTML
 - o Knapper ændret til Bootstrap
 - o Tabeller er blevet stribede
 - o Detalje visning af bil ændret til bootstrap card
 - o Default billede hvis dette ikke angives
 - o Visning af seneste 3 biler på Velkommen siden
 - o Detalje visning af bil ændret til et PartialView
 - o Udvidelse af client validering
- Performance optimering
 - o AsNoTracking
- Static files
 - o Brug af static files
- Azure
 - o Deploy af website til Azure

Opdatering af HTML

Knapper ændret til Bootstrap

Alle knapper i løsningen er blevet ændret til bootstrap udseende. Farvevalget følger for så vidt muligt navnene på bootstrap klasserne.



- Rediger = warning
- Detaljer = info
- Slet = danger

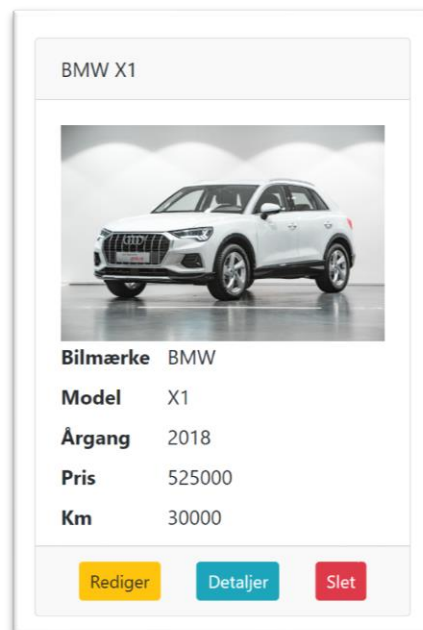
Tabeller er blevet stribede

De tabeller der er i løsningen er blevet stribede, for bedre at kunne skelne de enkelte rækker fra hinanden.

Bilmærke			
BMW	Rediger	Detaljer	Slet
Seat	Rediger	Detaljer	Slet
Skoda	Rediger	Detaljer	Slet
VW	Rediger	Detaljer	Slet

Detalje visning af bil ændret til bootstrap card

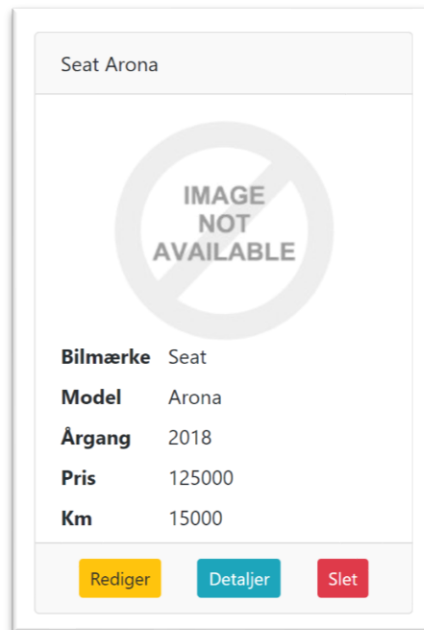
For at give et mere samlet overblik over hver enkelt bil, er informationerne nu samlet i et bootstrap card.



```
<div class="card mx-2 my-3 mt-lg-0">
  <div class="card-header">
    <span>@Html.DisplayFor(modelItem => Model.CarBrandName) @Html.DisplayFor(modelItem => Model.CarModelName)</span>
  </div>
  <div class="card-body pb-0">
    <div class="text-center">
      
    </div>
    <dl class="mb-0">
      <dt>@Html.DisplayNameFor(model => model.CarBrandName)</dt>
      <dd>@Html.DisplayFor(modelItem => modelItem.CarBrandName)</dd>
      <dt>@Html.DisplayNameFor(model => model.CarModelName)</dt>
      <dd>@Html.DisplayFor(modelItem => modelItem.CarModelName)</dd>
      <dt>@Html.DisplayNameFor(model => model.Year)</dt>
      <dd>@Html.DisplayFor(modelItem => modelItem.Year)</dd>
      <dt>@Html.DisplayNameFor(model => model.Price)</dt>
      <dd>@Html.DisplayFor(modelItem => modelItem.Price)</dd>
      <dt>@Html.DisplayNameFor(model => model.MileAge)</dt>
      <dd>@Html.DisplayFor(modelItem => modelItem.MileAge)</dd>
    </dl>
  </div>
</div>
```

Default billede hvis dette ikke angives

I tilfælde af der ikke er angivet et billede i url'en, vises et standardbillede.



```
public class CarsController : Controller
{
    private readonly ApplicationDbContext _context;
    private static string _NO_IMAGE;

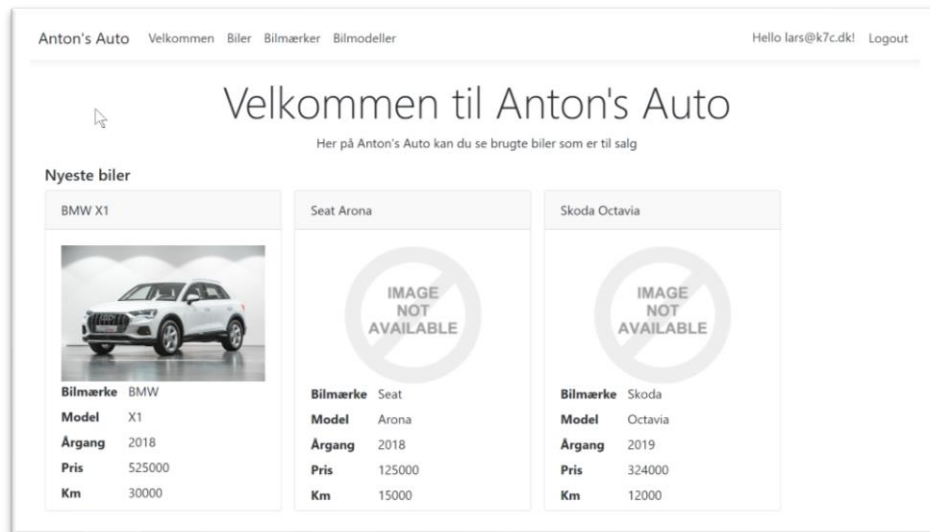
    public CarsController(ApplicationDbContext context)
    {
        _context = context;
    }

    public async Task<IActionResult> Index()
    {
        _NO_IMAGE = $"{HttpContext.Request.Scheme}://{HttpContext.Request.Host.Value}/images/image_not_available.jpg";
    }
}
```

Billedet hentes fra images folderen, som leveres via static files. Ud fra HttpContext'en hentes url scheme, base url, som sammensættes med stien til billedet.

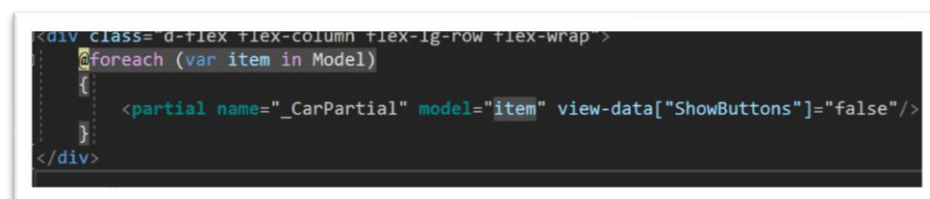
Visning af seneste 3 biler på Velkommen siden

På velkommen siden, som kan ses uden login, vises de 3 senest oprettede biler. Her er benyttet samme kort som på "Biler" siden



Detalje visning af bil ændret til et PartialView

For at kunne genbruge kortet for bil, er det lavet som et partialview. Dette gør at HTML koden hvor biler renderes minimeres, samtidig med at viewet kan bruges flere steder. Opdateres kortet vil det dermed slå igennem og give et ens udtryk de steder hvor en bil vises.



Ovenfor vises brugen af et partialview.

Udvidelse af client validering

For at give brugeren en bedre bruger oplevelse, er client side valideringen blevet udvidet med nogle flere regler.

```
public class CarViewModel
{
    [Key]
    public int CarID { get; set; }

    public int CarBrandID { get; set; }

    public string CarModelID { get; set; }

    [DisplayName("Bilmærke")]
    public string CarBrandName { get; set; }

    [DisplayName("Model")]
    public string CarModelName { get; set; }

    [Required(ErrorMessage = "Feltet er påkrævet")]
    [Range(1900, 2050, ErrorMessage = "Årgang skal være mellem 1900 og 2050")]
    [DisplayName("Årgang")]
    public int Year { get; set; }

    [Required(ErrorMessage = "Feltet er påkrævet")]
    [Range(0, 9999999, ErrorMessage = "Prisen skal være mellem 0 og 9.999.999")]
    [DisplayName("Pris")]
    public double Price { get; set; }

    [Required(ErrorMessage = "Feltet er påkrævet")]
    [Range(0, 9999999, ErrorMessage = "Antal Km skal være mellem 0 og 9.999.999")]
    [DisplayName("Km")]
    public int MileAge { get; set; }

    [DisplayName("Billede")]
    public string ImageUrl { get; set; }
}
```

Årgang har fået et område fra 1900 til 2050.

Pris og Km har fået et område fra 0 til 9999999.

Performance optimering

For at optimere performance i forbindelse med forespørgsler, hvor EF ikke skal holde styr på ændringer, kan der optimeres ved at benytte AsNoTracking. Dette er gjort på forespørgsler på index siderne

```
public async Task<IActionResult> Index()
{
    _NO_IMAGE = $"{HttpContext.Request.Scheme}://{HttpContext.Request.Host.Value}/images/image_not_available.jpg";

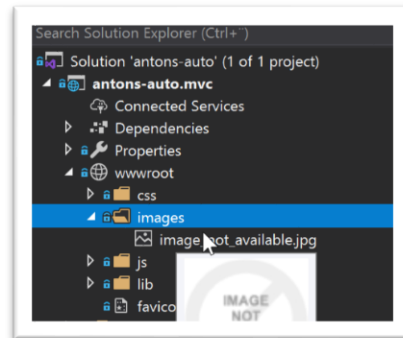
    var carsViewModel = await _context.Cars
        .Include(x => x.CarModel)
        .Include(x => x.CarModel.CarBrand)
        .OrderBy(o => o.CarModel.CarBrand.Name)
        .AsNoTracking()
        .Select(car => MapToViewModel(car))
        .ToListAsync();

    return View(carsViewModel);
}
```

Ovenfor er vist forespørgslen for biler siden.

Static files

Under wwwroot mappen er placeret filer som ASP.NET Core kan udstille. I denne mappe har jeg placeret det standardbillede, som skal vises når et billede url ikke er angivet.



Azure

Jeg har i denne uge arbejdet med at lære Azure at kende. I denne forbindelse har jeg oprettet et site og en database for Antons' Auto.

Anton's auto kan tilgås på linket

<https://sm-antons-auto.azurewebsites.net/>