# Opgave: Uge 5 – Anton’s Auto

Fag: Udviklingsmiljøer

Udarbejdet af: Lars Larsen

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

Github repository: <https://github.com/larsk7cdk/antons-auto.git>

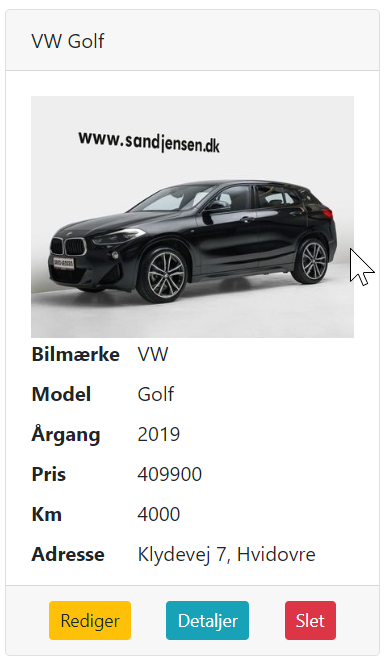
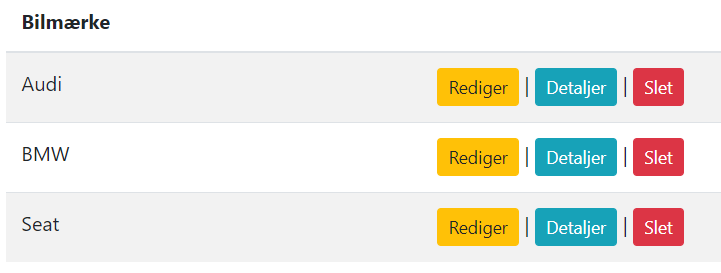
## I denne uge har jeg arbejdet med

* Opdatering af HTML
  + Knapper ændret til ”sm" Bootstrap
* Opret ny bil
  + Tilføjet adresse til en bil
* Google maps på detalje siden
  + Dawa (Danmarks Adressers Web API)
  + Google maps
* Sortering
  + Sortering på bilmodel siden
  + Sortering på bil siden
* Søgning
  + Søgning på bilmærke og bilmodel på bil siden
* Sideinddeling på alle bil sider
  + Forrige og Næste knapper på siden
* ”Om” Side
  + Karrusel til visning af biler

## Opdatering af HTML

Knapper ændret til ”sm” Bootstrap

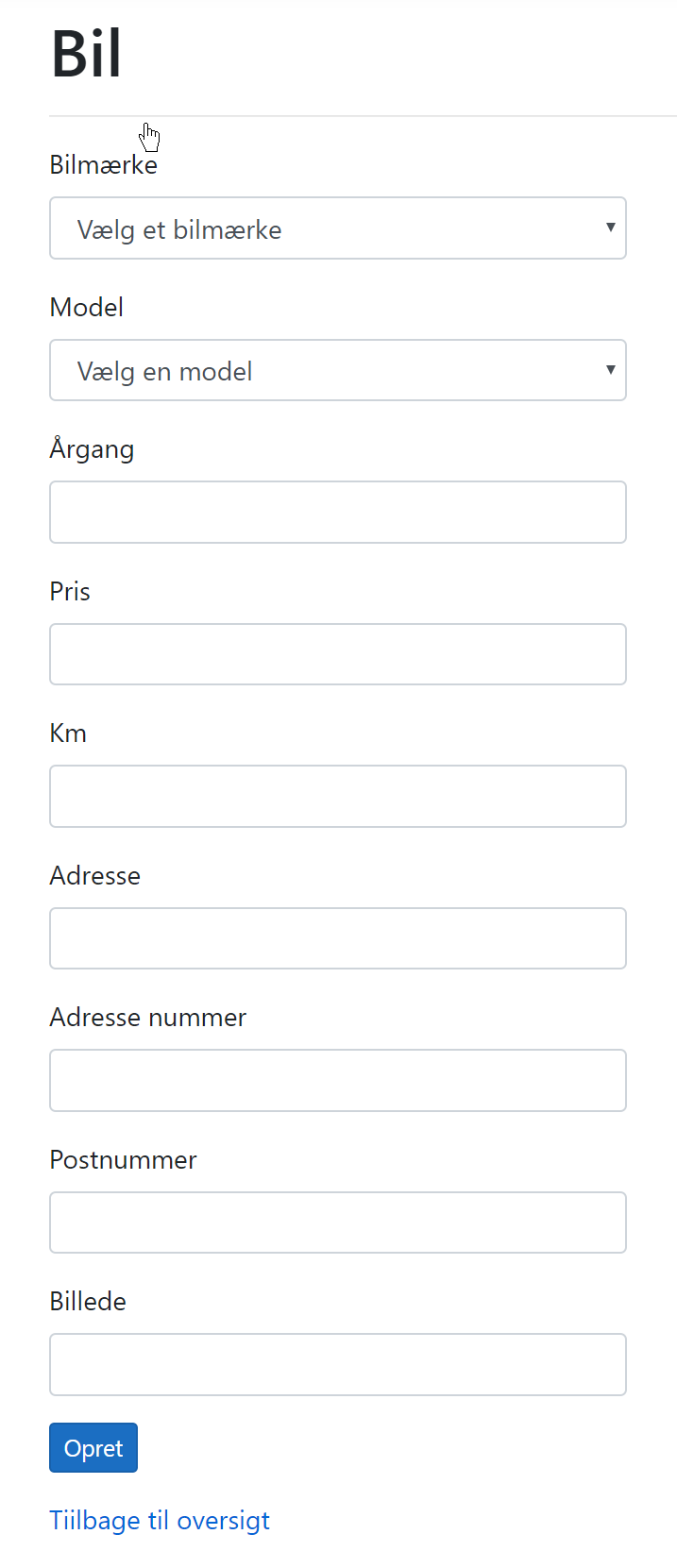
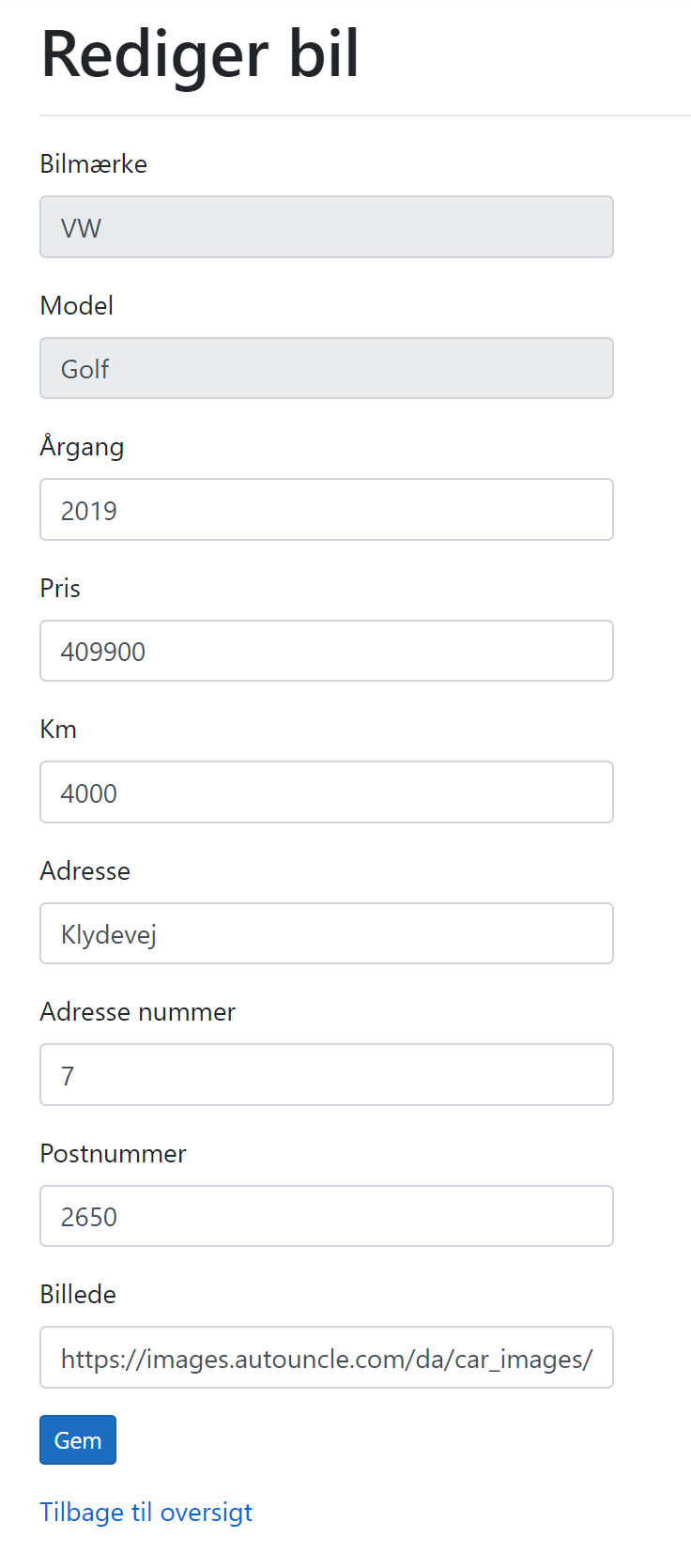
Knapper på bil kort og tabeller er ændret til en ”btn-sm” klasse. Dette gør knapperne virker mindre dominerende



## Opret ny bil

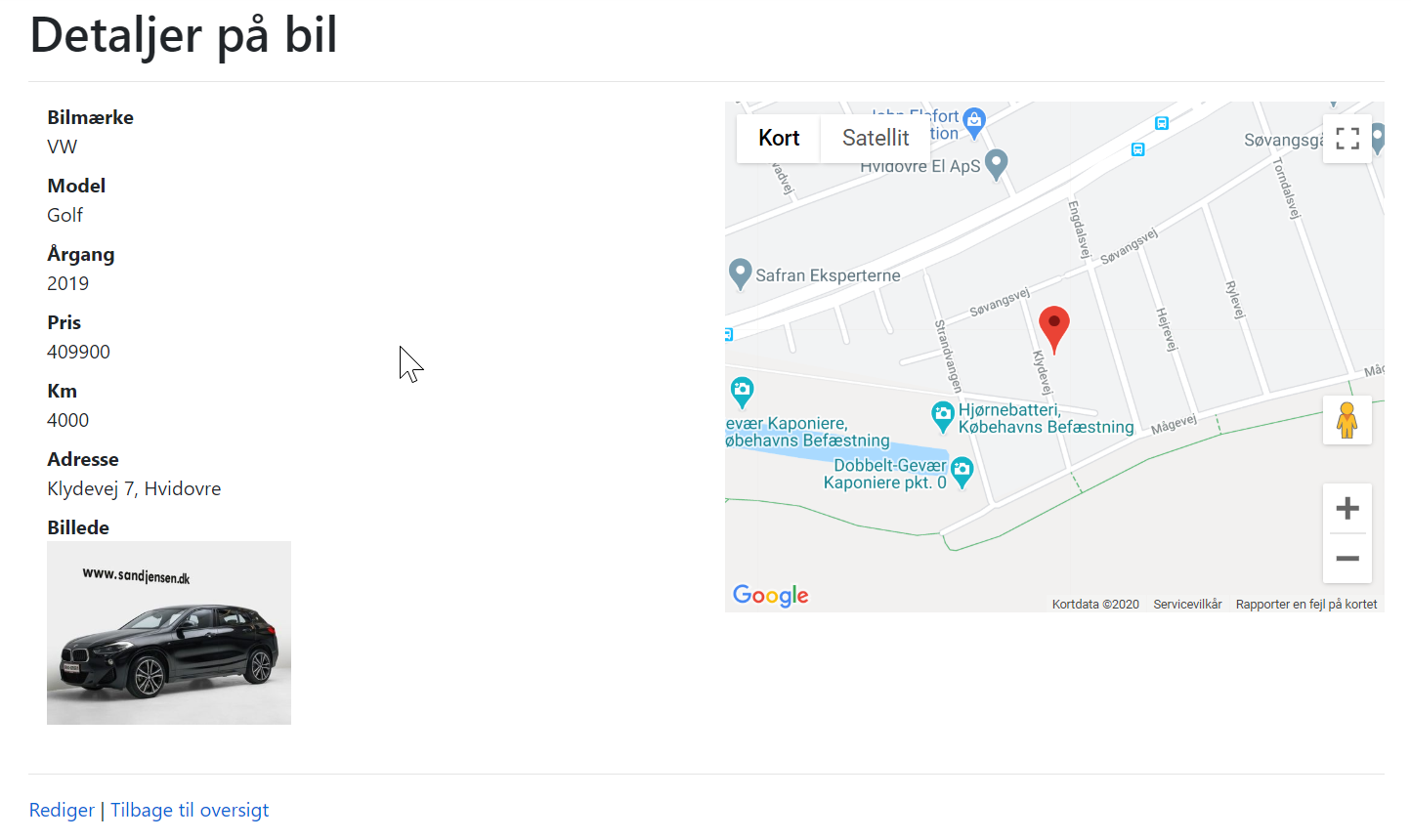
Tilføjet adresse til en bil

Når man opretter en ny bil, er der nu felter hvor adressen kan angives. Oplysningerne gemmes i databasen og kan senere rettes hvis det er nødvendigt på redigerings siden

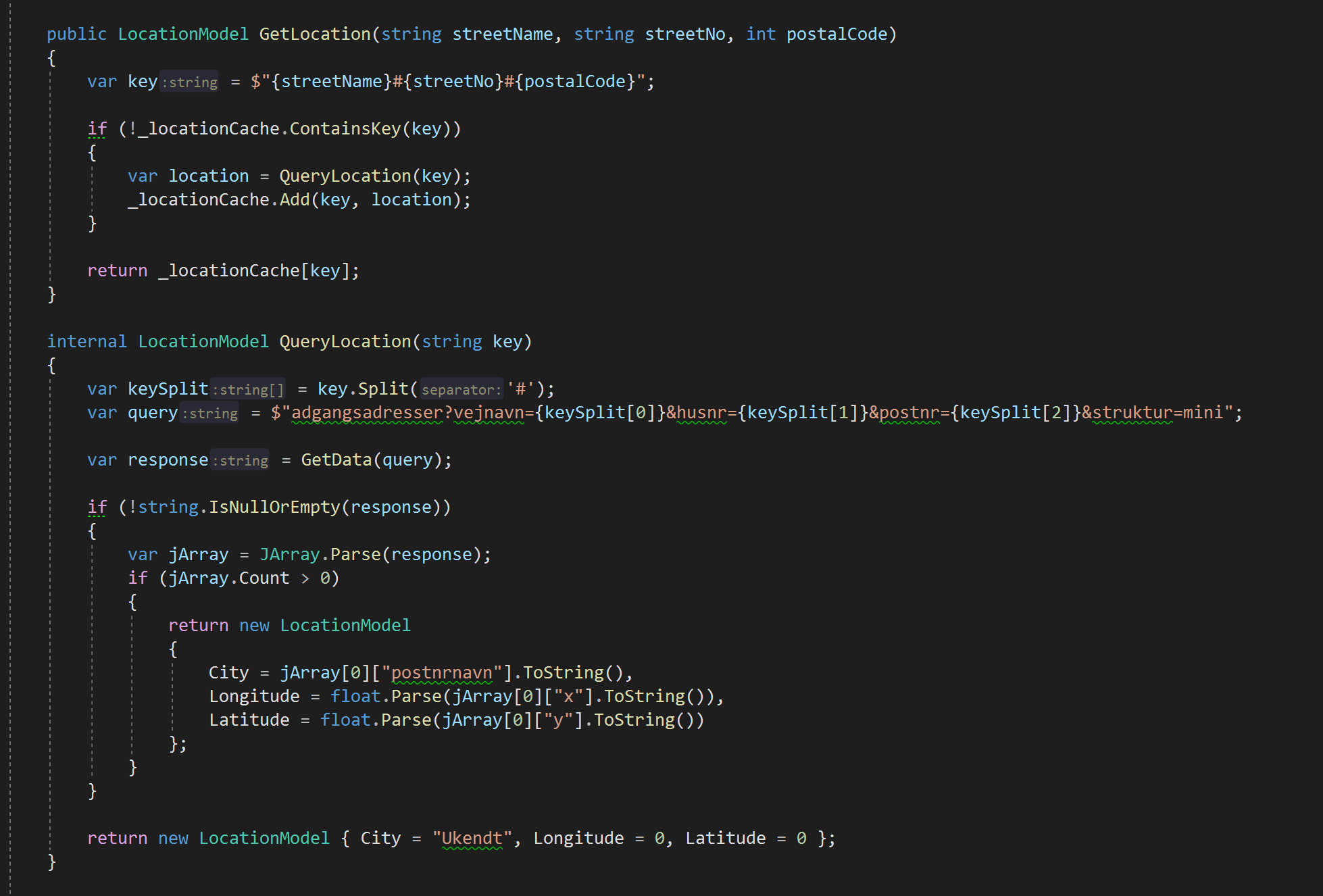
## Google maps på detalje siden

Hvis der er registreret en adresse på en bil, og man går ind på detalje siden, vises dette på et Google maps kort



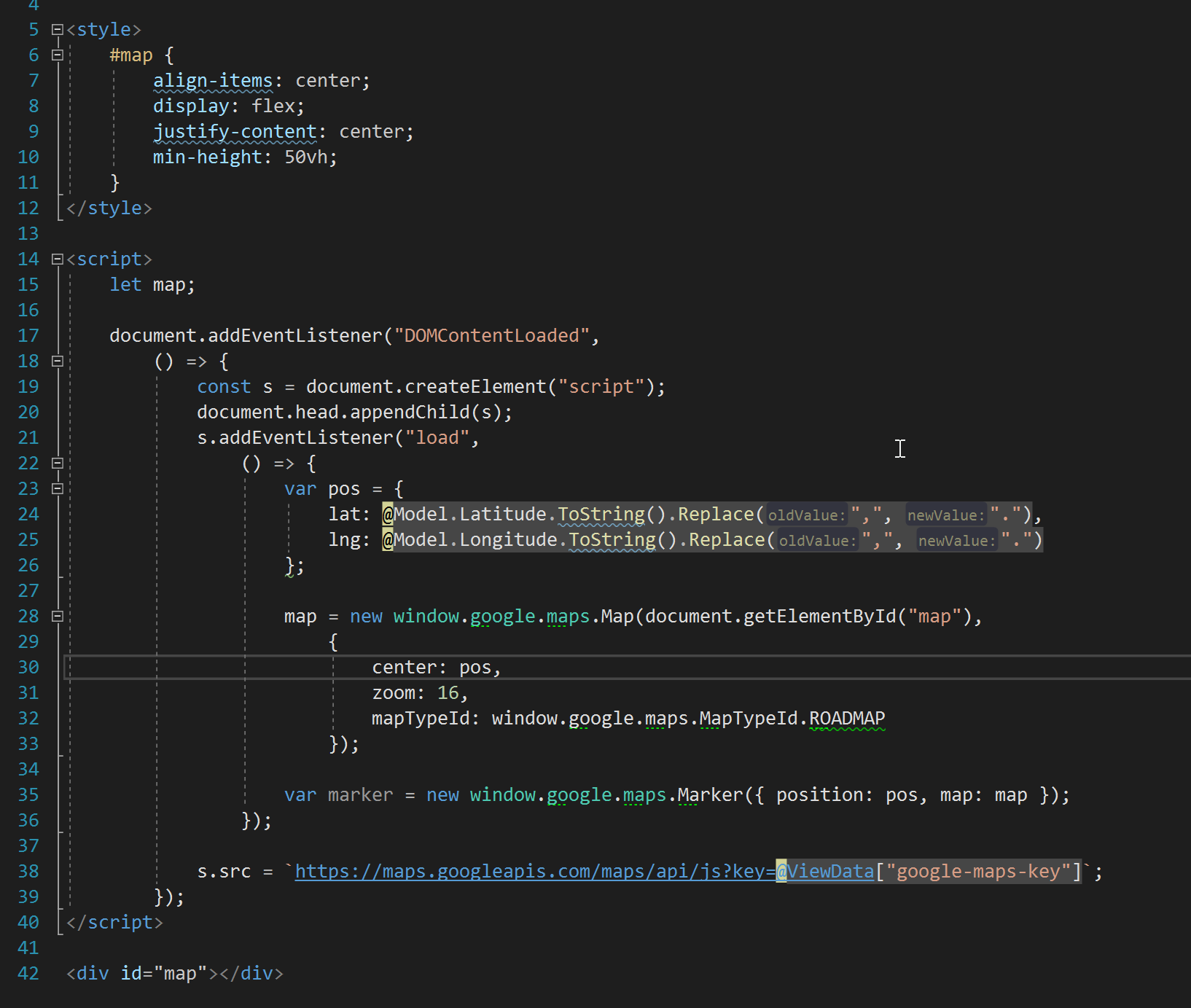
Dawa (Danmarks Adressers Web API)

For at finde længde- og breddegrader for en angivet adresse, er der lavet en serviceproxy, som kan hente disse fra Dawa. Nedenstående viser et udsnit af DawaServiceProxy. Den har en metode der kaldes med vejnavn, vejnummer og postnummer. Ud fra de oplysninger hentes længde-og breddegrader til brug for Google maps



Google maps

Google maps er lavet som et partial view og bliver vist ved brug af nedenstående javascript kode. Når siden med dette view renderes, tilføjes map’et til DOM’en og på de givne længde- og breddegrader sættes en markør

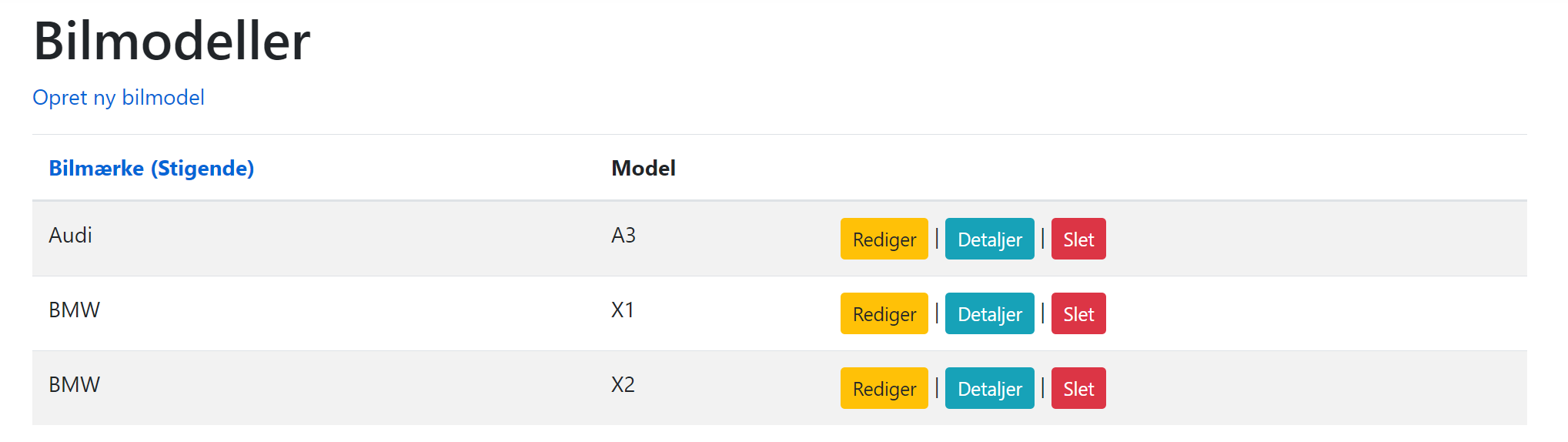


## Sortering

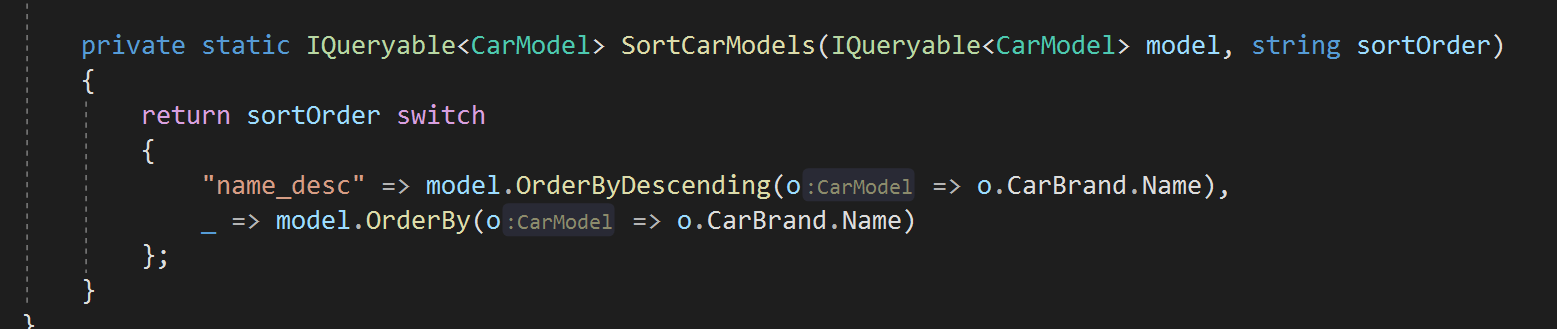
Der er lavet sortering på bilmodel og bil siden.

Sortering på bilmodel siden

På bilmodel siden er sorteringen lavet ved klik på tabellens bilmærke overskrift. Ved klik skifter sortering i henholdsvis stigende og faldende sortering

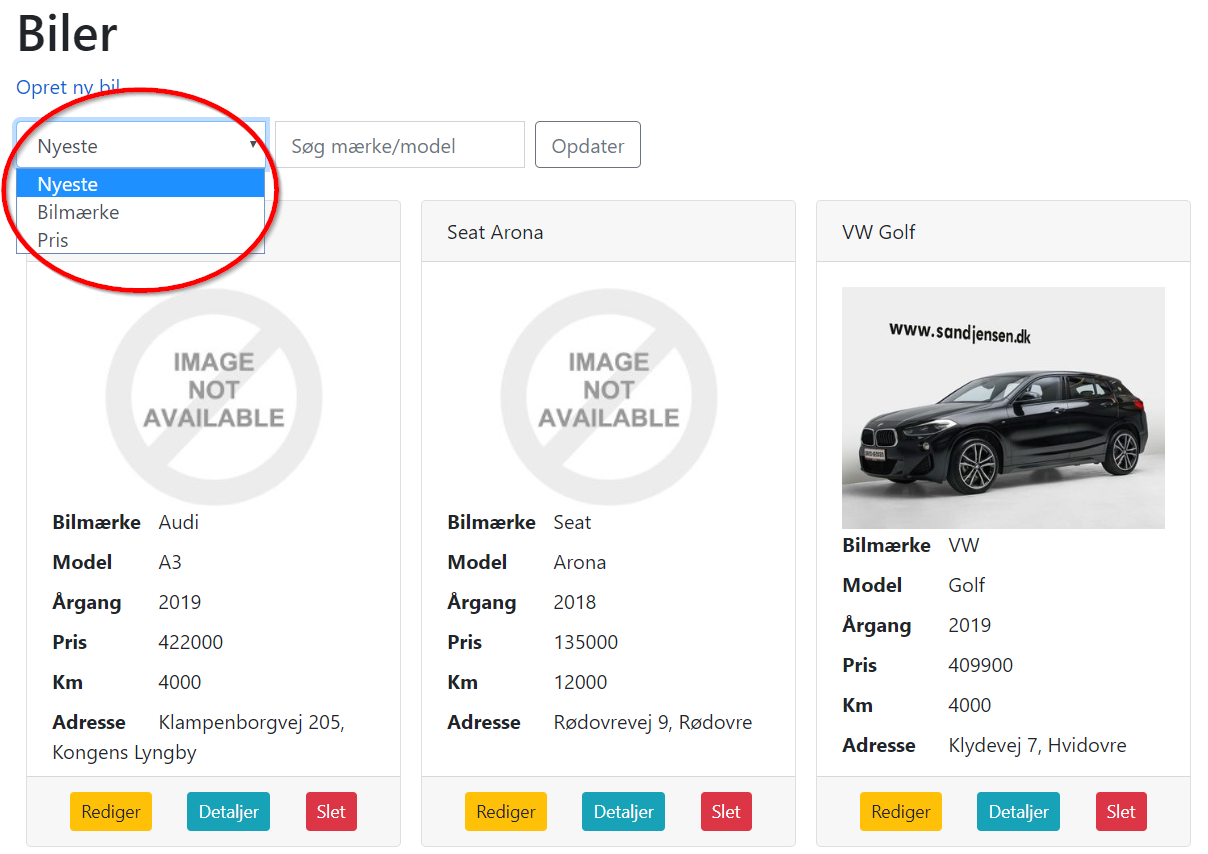


Sortering foretages ved at udvide LINQ udtrykket, som henter bilmodeller med nedenstående kode

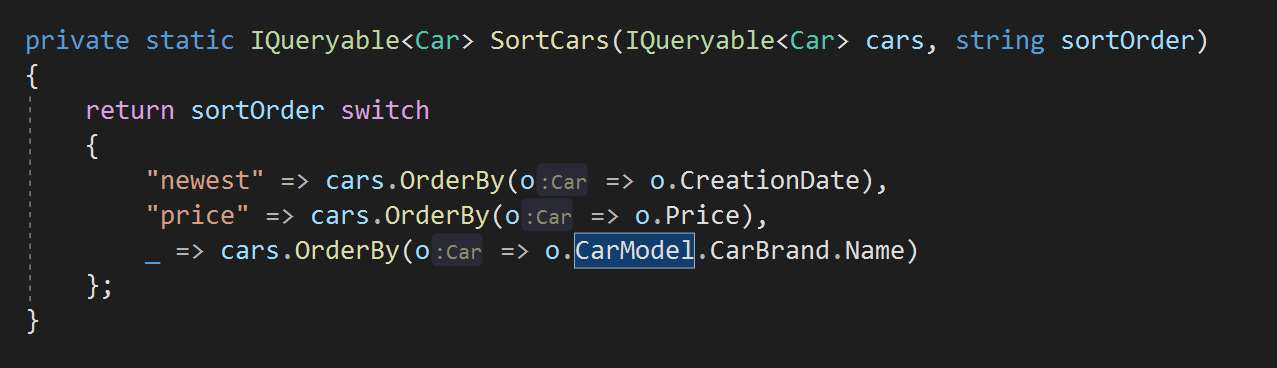


Sortering på bil siden

På bil siden er sortering lavet ved brug af en dropdown liste. Når en given sortering er valgt opdateres ved klik på ”Opdater” knappen



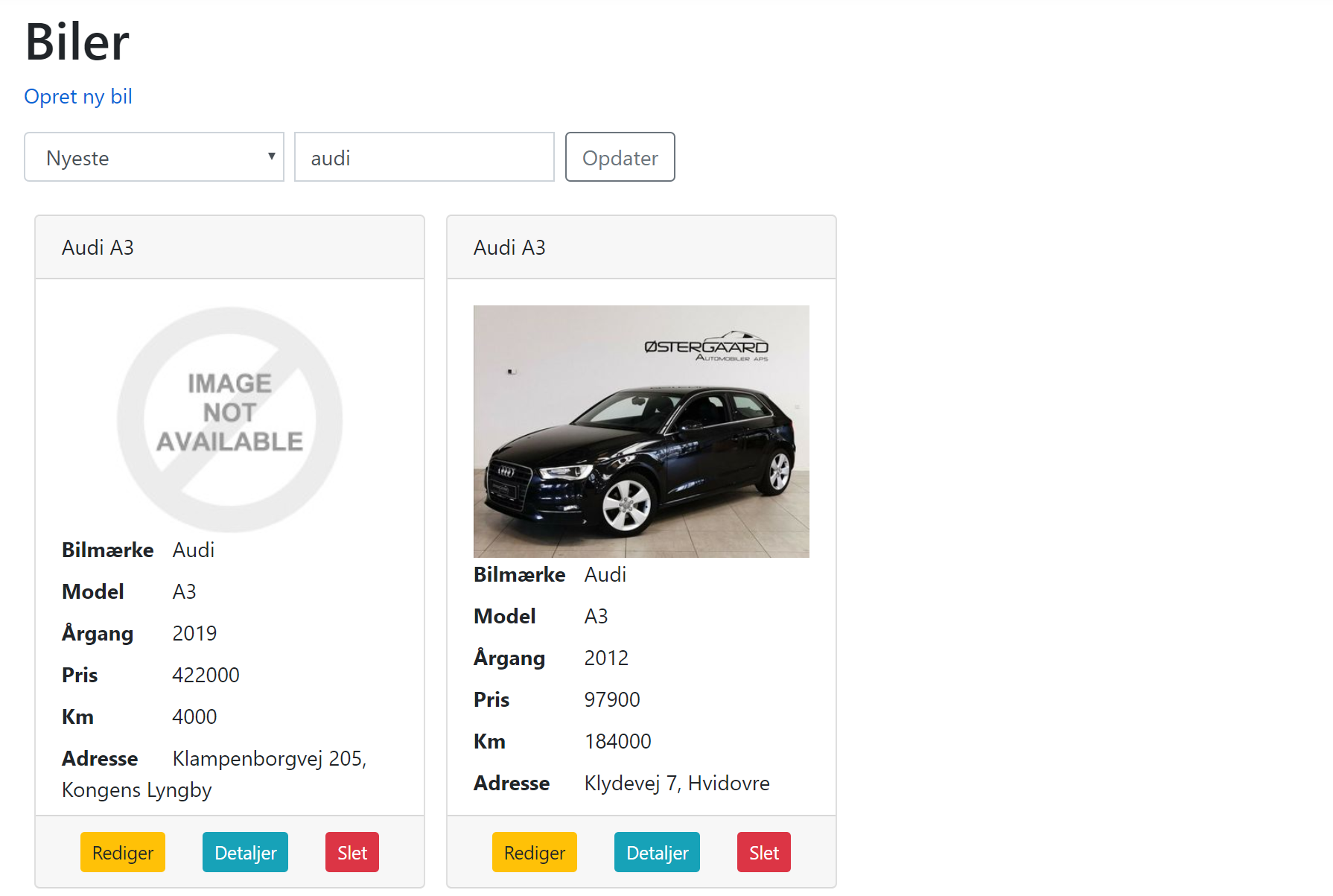
Sortering foretages ved at udvide LINQ udtrykket, som henter biler med nedenstående kode



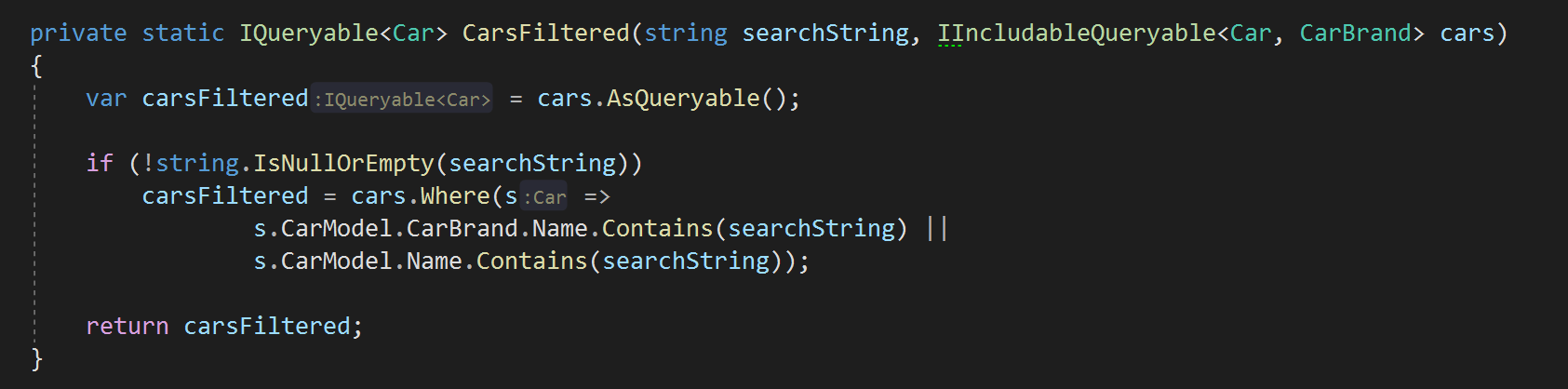
## Søgning

Søgning på bilmærke og bilmodel

På bil siden er det muligt at indtaste en tekst, som laver en søgning på bilmærke og bilmodel. På nedenstående vises en søgning på audi

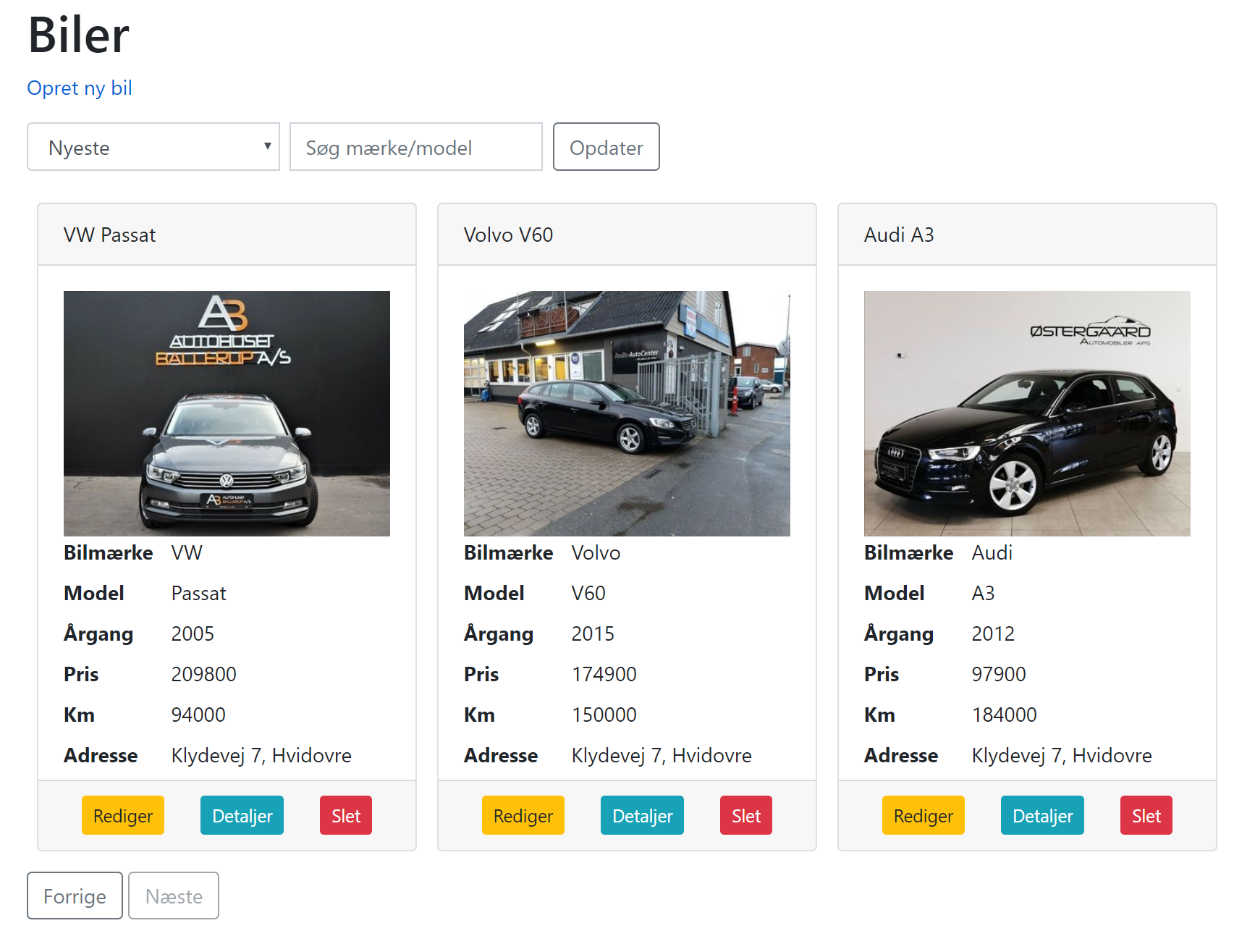


Søgning foretages ved at udvide LINQ udtrykket, som henter biler med nedenstående kode



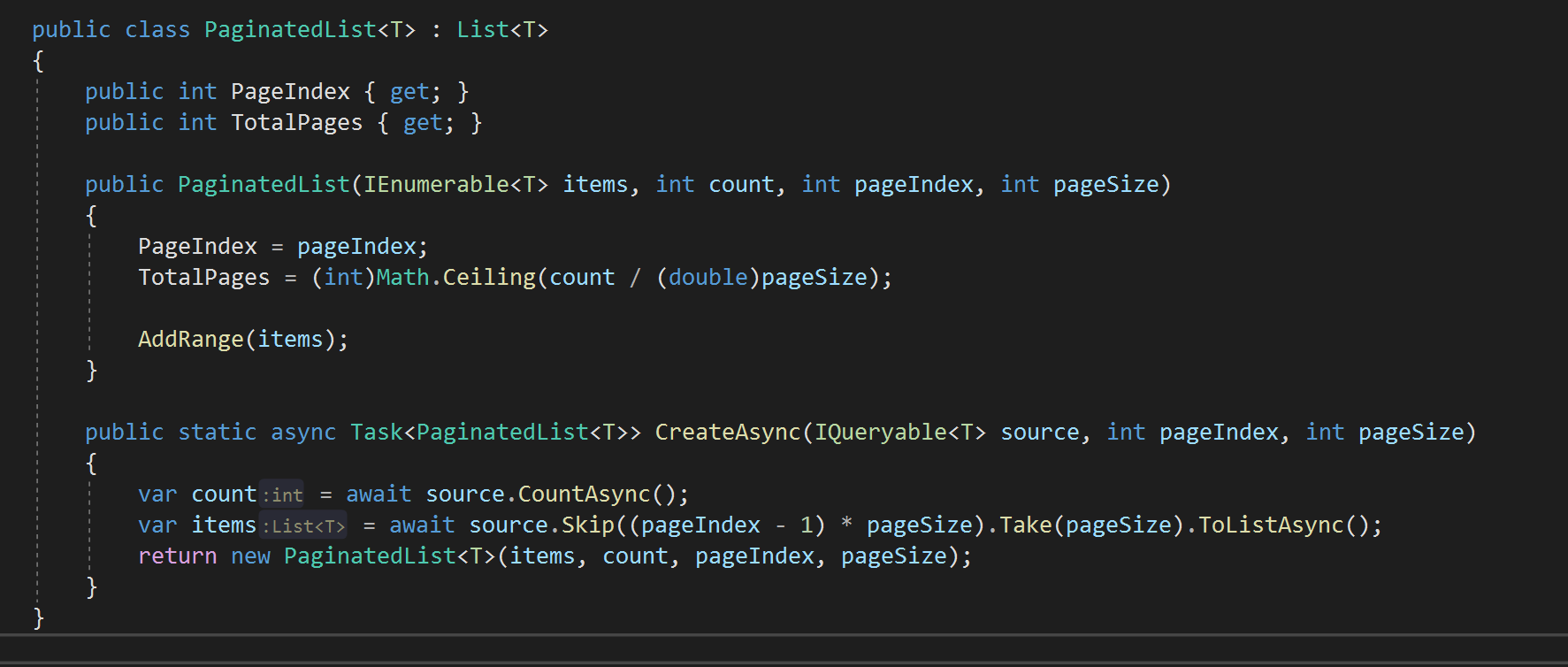
## Sideinddeling på alle bil sider

På bilmærke og bilmodel siden er der lavet en sideinddeling så der vises 3 poster pr. side. På bil siden er antallet sat til 6. Her vises den sidste side hvor ”Næste” knappen er disabled, fordi der ikke er flere sider at vise

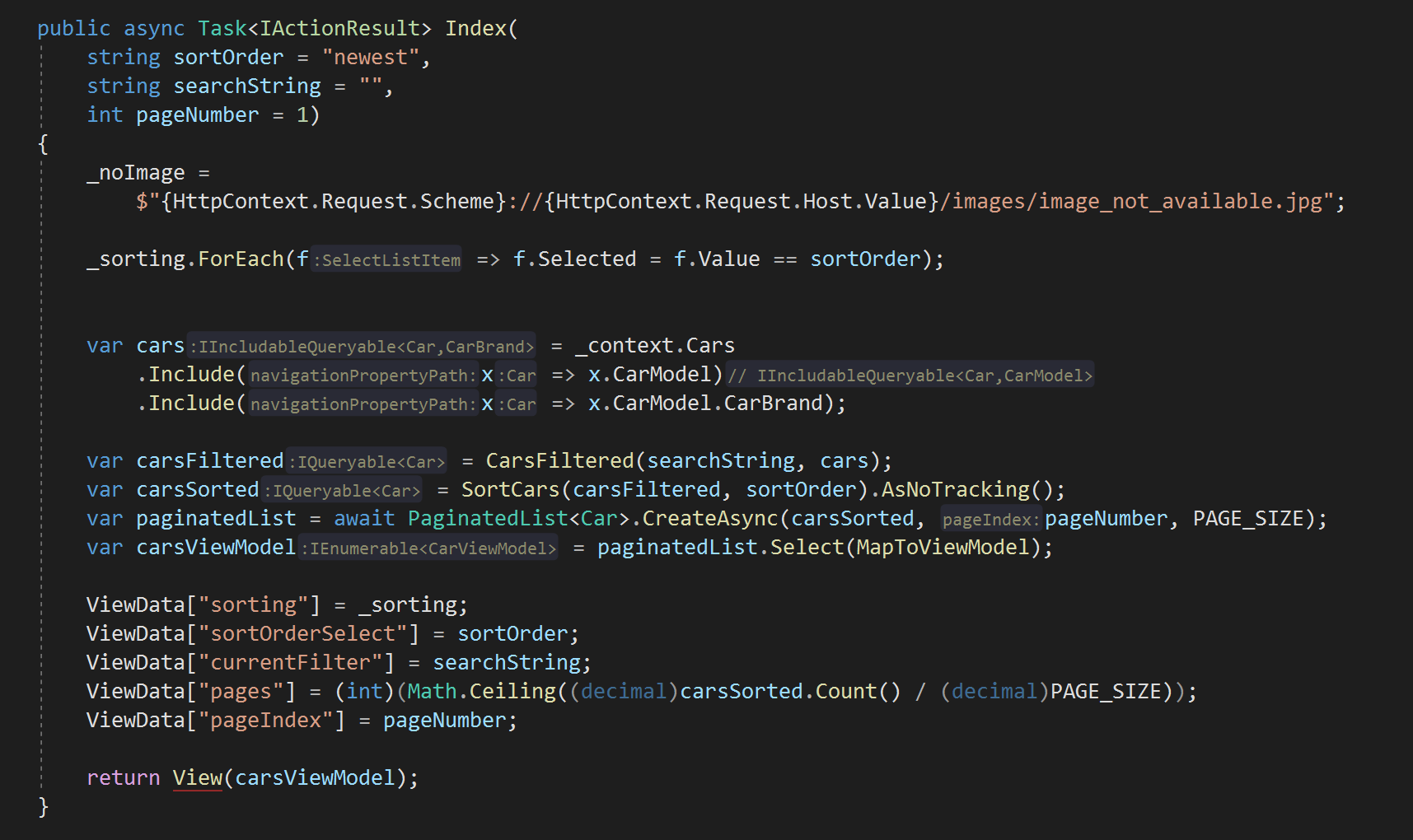
**

PaginatedList

Nedenstående hjælpeklasse bruges til at lave sideinddeling af poster



Her vises koden for index klassen af bil siden. ViewData bruges til at sende data mellem klient og server siden.



Forrige og Næste knapper på siden

”Forrige” og ”Næste” knapperne gør brug af et anchor tag. For at bibeholde de valgte værdier for sortering og søgning, sendes de med når der skiftes side



## ”Om” side

Karrusel til visning af biler

På ”Om” siden har jeg lavet en karrusel, som viser de biler der har en imageUrl



Til dette formål har jeg benyttet Bootstrap. Ved første gennemløb skal ”active” klassen sættes på det første element

