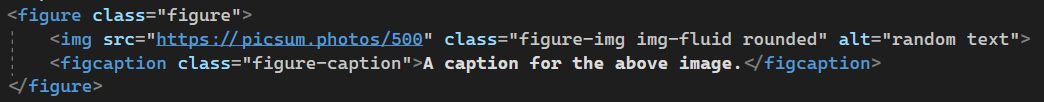
Functional Requirements:

Given a list of Cars, you will need to implement 2 pages:

First Page:

1. Contains two different ways to list the cars
   1. Using a table.
   2. Using a list.
2. Each car entity should have (Manufacturer, Model, Image, HtmlDescription, FirstUseDate)
3. Display the name of the cars as ({Model} From {Manufacturer}).
4. Display the usage duration (Seek for help for how to deal with DateTime).
5. Display the HtmlDescription as rendered Html not as plain text.
6. A button that redirects the user to a single element view.
7. Image should be rendered inside a figure tag and the alt attribute and the figcaption tag should contain the name of the car ({Model} From {Manufacturer}).  
   Ex:  
   
8. A button that lets the user choose the rendering method.

Second Page:

1. Details Page
2. Contains the details of the element and message to tell whether the user came from a list or a table.

Technical Requirements:

1. Reuse your code as much as possible and reduce duplicate code.
2. Use razor view engine features.
3. Use Render Methods.
4. Use query string parameters.
5. Both pages can share the same view.

Tips:

1. Make the pages share the same view.
2. Make the pages receive the same model type.
3. Create an enum that contains three rendering options (Table, List, LetUserChoose) and use.
4. The model type passed to the view should contain two properties:
   1. List of cars.
   2. Rendering option.

Bonus:

Do your best and you’ll get a bonus on how much you achieved DRY principle.