# Exercise: Razor Engine & Filters

Problems for exercises and homework for the [“C# MVC Frameworks - ASP.NET Core” course @ SoftUni](https://softuni.bg/trainings/1797/c-sharp-mvc-frameworks-asp-net-core-october2017).

## Camera Bazaar

You need to create an **online store** for shop selling digital cameras. All visitors of the store should have access to the **list of cameras** and could see their **details**. All visitors could **register** into the store. The registered users can **add, edit** and **delete** cameras. Each user can **modify** only those cameras that are **added by him**.

The requirements for the **models** are:

* **User**
  + **Username** – must be between **4** and **20** symbols long and must contain **only letters**. Casing **does not** matter.
  + **Email** – must be a **valid email address**
  + **Password** – must be at least **3 symbols** long and can contain only **lowercase letters** and **digits**
  + **Phone** – must start with “**+**” sign followed by **10 to 12 digits**
* **Camera**
  + **Make** – makes can be only **Canon**, **Nikon**, **Penta** or **Sony**
  + **Model** – can contain **only uppercase letters**, **digits** and **dash** (“**-**“). Cannot be empty
  + **Price** – floating point number with precision to **2 digits** **after** floating point
  + **Quantity** – integer number in range **0 – 100**
  + **Min shutter speed** – integer number in range **1 to 30** (seconds)
  + **Max shutter speed** – integer number in range **2000 to 8000** (fraction of a second)
  + **Min ISO** – can be **50 or 100**
  + **Max ISO** – integer number in range **200 to 409600** and must be **dividable by 100**
  + **Is Full Frame** – **yes** or **no**
  + **Video Resolution** – described with text **no longer than 15 symbols**
  + **Light Metering** – can be **spot**, **center-weighted** and **evaluative**
  + **Description** – details for the camera with no more than **6000 symbols**
  + **Image URL** – must start with **http://** or **https://**

In each form that **add** or **edit data** if there is a mistake anywhere **fill** the **form** fields from the model, **notify the user** and provide him information **what exactly is the mistake** and why some data cannot be accepted.

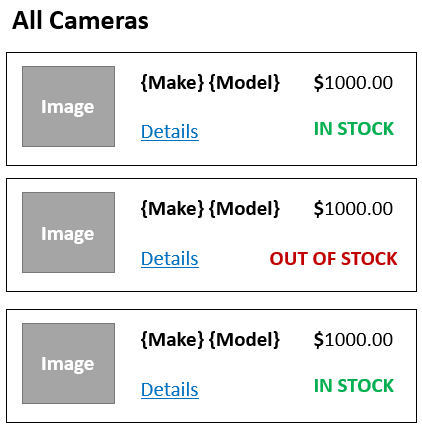
**Hint:** Use **HTML/Tag** Helpers

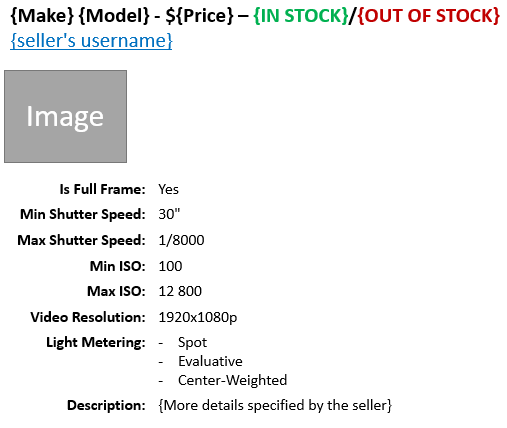
Each **user** should have **profile page** that contains information about **him, list of all his cameras** and summarized count of in stock or out of stock cameras. Profile pages should be accessible **only by registered users**. Each user can **change** his **email,** **phone** or **password**.

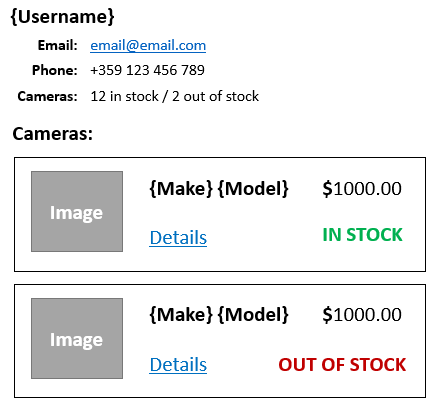
Each **camera** should have page where **all details** about it can be seen as well as **link** to the profile to the user who added it.

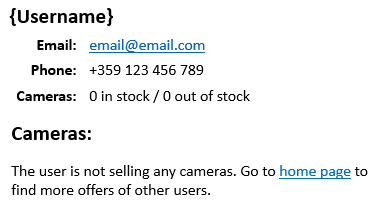
On the **list of cameras**, the visitors should be able to see clearly whether the **camera is** **in stock or not**. For example, if it in stock add green text “IN STOCK” otherwise red text “OUT OF STOCK”.

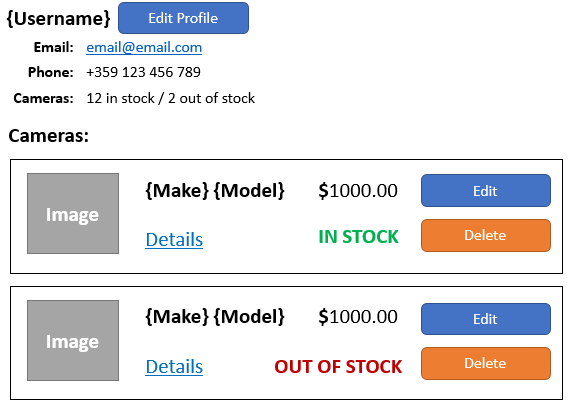


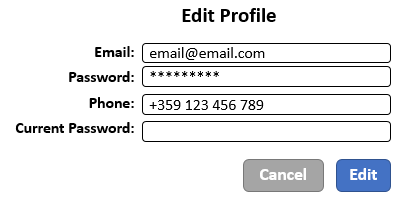












## Log Filter

Create custom action **filter** that **logs action when it is performed**. For each log, you should keep information about:

* **Date and time** when the action method was executed
* **IP address** of the user who sends the request
* **User** – name of the user who executed that action or if the user is not logged in print “anonymous”
* **Controller** and **action**
* **Exception** **type** and **message** if there is any

Use the following format for each log.

* Log after **successful execution** of an action:

{date and time} – {IP Address} – {User/Anonymous} – {Controller}.{action}

* Log format when **exception** is thrown:

[!] {date and time} – {IP Address} – {User/Anonymous} – {Controller}.{action} - {Exception type} – {Exception Message}

Logs should be saved in a file **logs.txt.**

Try logging only execution of **some actions**. Then try to log execution of **all actions** in given controller. Finally set the logging **globally** to the whole application.

### Example

|  |
| --- |
| **logs.txt** |
| 2017-03-02 10:00:00 – 192.168.0.1 – Anonymous – HomeController.Index  2017-03-02 10:13:12 – ::1 – Anonymous – HomeController.Contacts  2017-03-02 12:10:00 – 128.128.128.14 – Anonymous – AccountsController.Login  2017-03-02 12:10:20 – 128.128.128.14 – Dragon123 – AccountsController.Edit  2017-03-02 12:11:20 – 128.128.128.14 – Dragon123 – AccountsController.ChangePassword  2017-03-02 12:12:20 – 128.123.192.16 – Pesho – HomeController.Index  [!] 2017-03-02 14:00:00 – ::1 – Anonymous – AccountsController.Edit – UnauthenticatedException – You must be logged in to access the page |