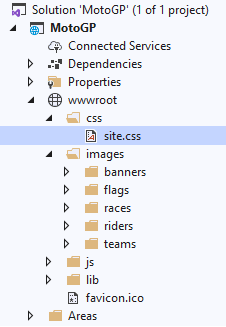
***MotoGP – part 1***

In this exercise, we will create a website for *MotoGP*, the Formula 1 motorbike circuit. In the first part of the exercise, we will create the project, adjust the default template and create quite a few empty (for the moment at least) controllers and views.

In the folder *www*, create a new *ASP.NET Core Web App (Model-View-Controller)* with the name *MotoGP*. Use *.NET 7.0 and Individual Accounts*.

To develop the website you will need a number of images. Copy the *images* folder (containing the folders *banners*, *flags*, *races*, *riders* and *teams*) to the *wwwroot* folder:



Also overwrite *site.css* in the *css* folder with the *site.css* of the *MotoGP* resources.

***1. \_NewLayout.cshtml***

Create an extra *layout template page* named *\_NewLayout.cshtml*. The content is



Note there's a link to the site's *css*. The *~* is a shortcut to the *wwwroot* directory.

***2. Home Page***

We will now modify the *Index* view of the *Home* controller. Get rid of all the code in it and replace it with the following code:

@{

Layout = "~/Views/Shared/\_NewLayout.cshtml";

ViewData["Title"] = "Home Page";

}

<style>

.centered {

position: fixed;

top: 50%;

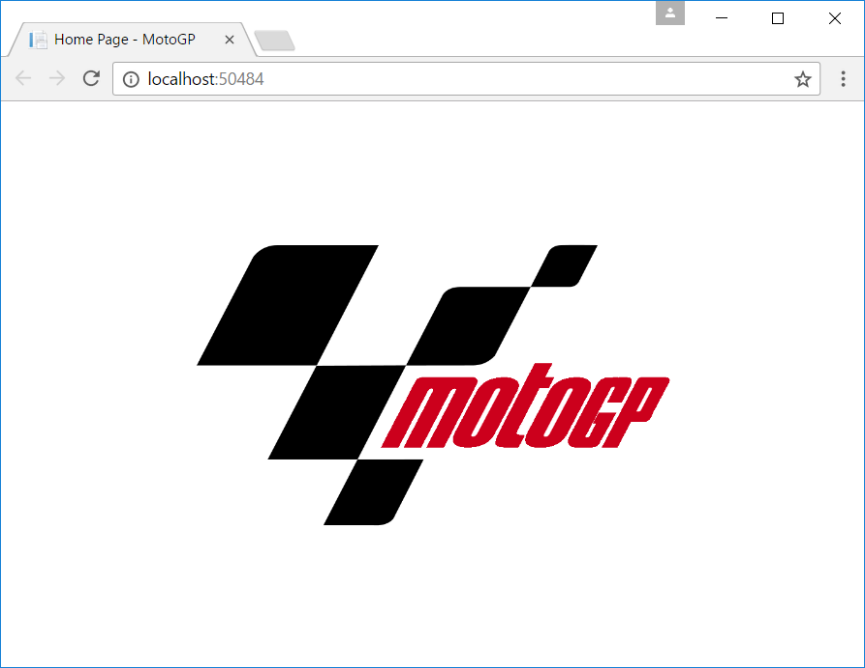
left: 50%;

transform: translate(-50%, -50%);

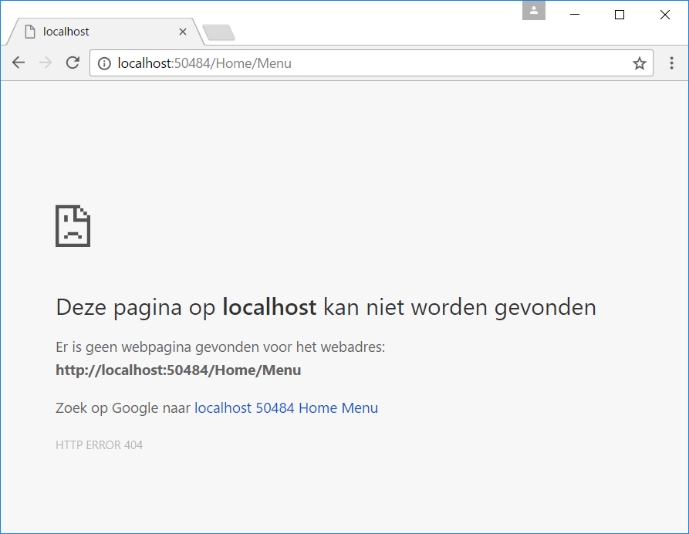
}

</style>

The first line changes the layout template for this page to the template we've just created. We also set the *Title* of the page. An inline-style a class *.centered* is defined. Use this class to center the *MotoGP* image. Complete the *Index view* yourself (after pressing *Ctrl-F5*):



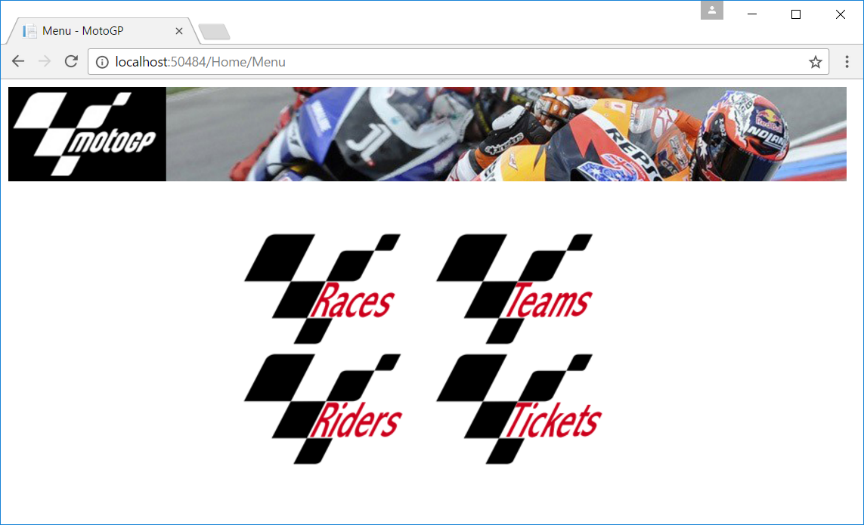
You see a centred image with the *MotoGP* logo. The width of the image is 400. Use a tag helper to turn that image into a hyperlink. If you click on it, jump to the *Home* controller's *Menu* method (which does not exist yet):



***3. Main Menu***

Create an additional method *Menu* in the *Home* controller. Generate in the method a random number between 1 and 4 (both inclusive). Put this number in the *DataView* dictionary with key *BannerNr*. Then call the view.

Now create the view. This view also uses *\_NewLayout.cshtml* as layout template:



At the top, you can see a banner. Which banner is shown, is decided based upon the value of the random number in the *DataView* dictionary. So, when you request the page several times in a row, you will get to see a different banner each time (since the number is random). All banners (*banner1.jpg*, *banner2.jpg*, *banner3.jpg* and *banner4.jpg*) can be found in the *images/banners* folder.

The main menu is made up by the four images. The following table shows the names of the images and the *Controllers/Methods* you jump to when you click on them (use a **tag helper**):

|  |  |
| --- | --- |
| ***image*** | ***link to*** |
| ~/images/banners/races.png | *Info/ListRaces* |
| ~/images/banners/teams.png | *Info/ListTeams* |
| ~/images/banners/riders.png | *Info/ListRiders* |
| ~/images/banners/tickets.png | *Shop/OrderTicket* |

Try to arrange these four hyperlinks nicely in the middle of the page. Use your *css* knowledge for this.

In the title tab, show *Menu - MotoGP*.

***4. \_Layout.cshtml***

We will now change the *default layout page*. Remove all code and copy the code from *\_NewLayout.cshtml* to *\_Layout.cshtml*.

Now define a variable *banners* in *\_Layout.cshtml* with all possible banners that can be used in the application (these are not the same banners as the one we used in the *Home Page*):

@{

var banners = new List<String>

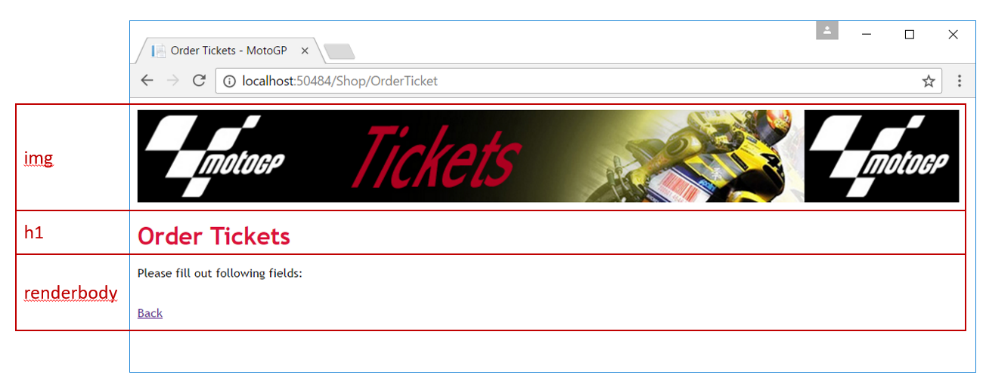
{ "bannerRaces.jpg", "bannerRiders.jpg", "bannerTeams.jpg", "bannerTickets.jpg" };

}

<!DOCTYPE html>

In the *body* part, you put an *img*. Which image is shown in that *img*, depends on the value of *BannerNr* that you get from the *ViewData* dictionary (since there are four banners in the list, *BannerNr* will be a value between 0 and 3). Under the banner, you show an *h1* containing the *Title* (also from *ViewData*) of the page.

This is the result if *BannerNr* has the value of 3 (you can only test the template later when you also create the views):



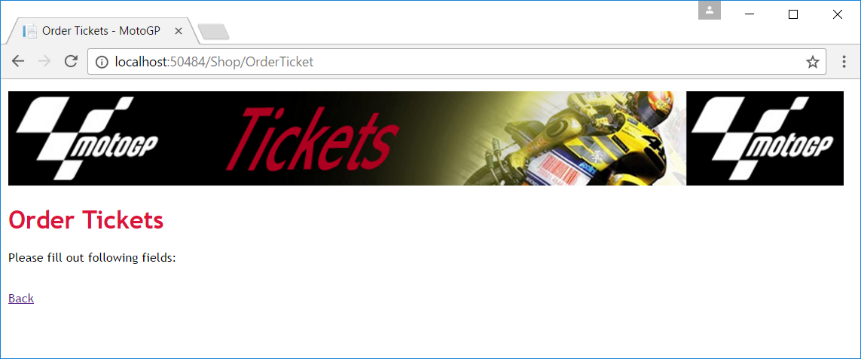
Make sure you can click on the banner. Jump back to the homepage.

***5. Order Tickets***

Through the application, tickets can be ordered for the *MotoGP* races. These reservations are of course written into a database (we'll do this later in the exercise). For now, we'll just create the controller and the views.

Create a new controller *Shop* (respect the naming convention for controllers). Create an *OrderTicket* method within it (you can discard the *Index* method). Set *BannerNr* to 3 and call the view.

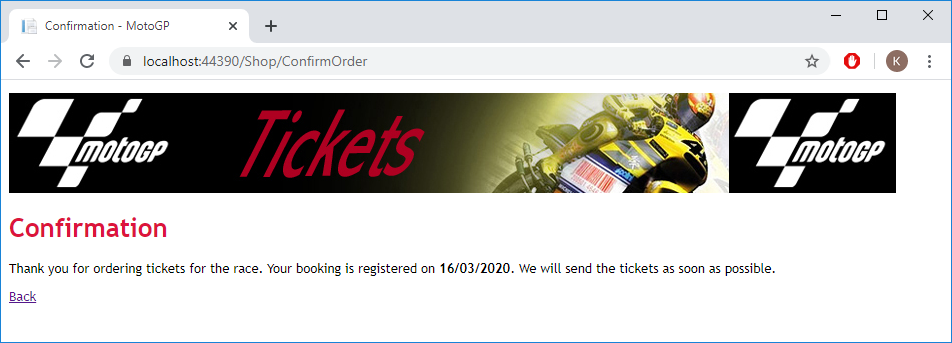
Now create the view. This is the result:



*Back* takes you back to the menu.

***6. Confirmation***

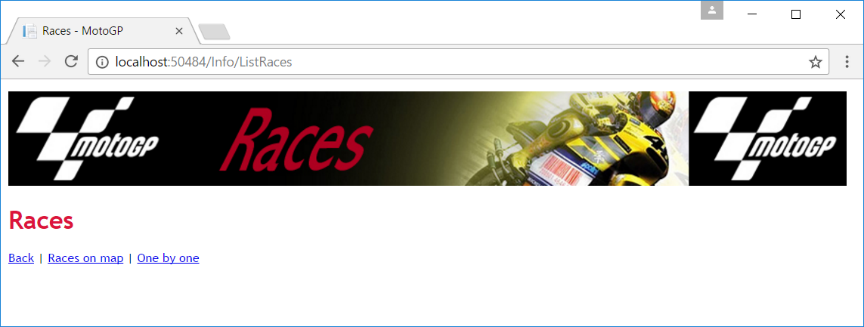
Create an additional *ConfirmOrder* method in the *Shop* controller. Set *BannerNr* to 3 and call the view. Create the view (test it via the address bar for now):



This page will later be used to show a confirmation when a visitor orders tickets. Note that in the confirmation text the *current date* is used in the format *dd/mm/yyyy*. Clicking the hyperlink takes you back to the main menu.

***7. List Races***

Create an additional controller *Info*. Create a method *ListRaces* with an associated view. The result is (for now):



Make sure you use the right *banner* (number 0) and *title*. Below the title you will see three hyperlinks. *Races on* *map* jumps to *Info/BuildMap* and *One by one* to *Info/SelectRace*.

***8. Race class***

Create an additional class *Race.cs* in the *Models* folder. Using the abbreviated *CSharp* notation, give this class the following properties:

public class Race

{

public int RaceID { get; set; }

public string Name { get; set; }

public int X { get; set; }

public int Y { get; set; }

}

***9. Races on map***

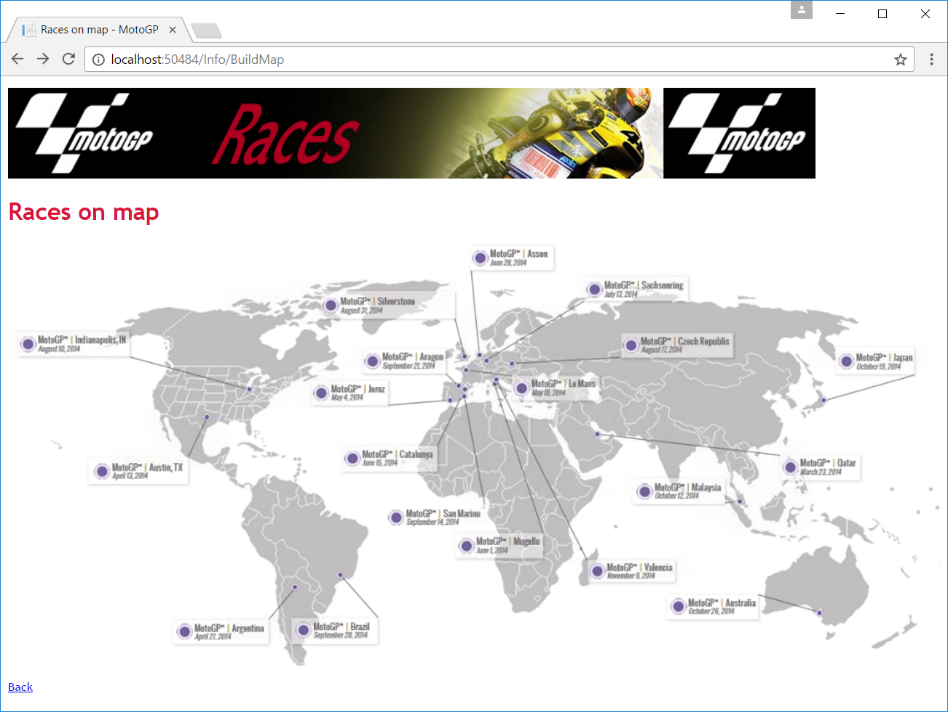
In the *Info*-Controller, create an additional method *BuildMap* and its associated view. Before calling the view, create a *List* of following *Race* objects (hardcoded for now):

{ RaceID = 1, X = 517, Y = 19, Name = "Assen" }

{ RaceID = 2, X = 859, Y = 249, Name = "Losail Circuit" }

{ RaceID = 3, X = 194, Y = 428, Name = "Autódromo Termas de Río Hondo" }

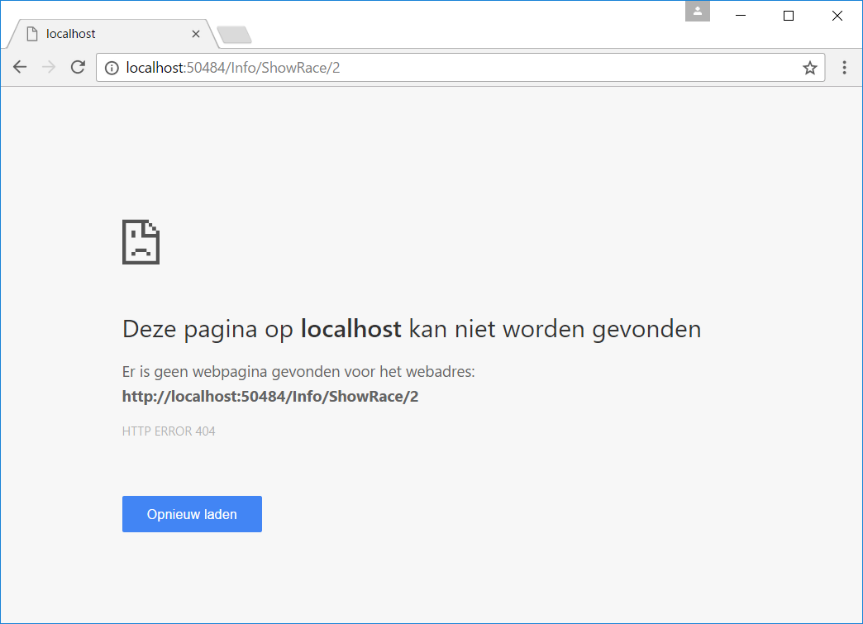
Pass those races (besides *BannerNr*) to the view. In that view, you show an image with all the *MotoGP* races in the world (*map.png*):



Now make this map *clickable* (more info on <http://www.w3schools.com/tags/tag_area.asp>). For each race in the *List* (there are three for now) create a circle which you can click on. If you click on a race, jump to the *ShowRace* method of the *Info* controller. The *id* of the race is the first parameter (use the *url helper*). For example for the *Losail Circuit* in *Qatar*:

/Info/ShowRace/2

For the time being, you will receive the following error message because the *ShowRace* method does not yet exist:



***10. Test***

Quickly try out all the hyperlinks from the homepage:

