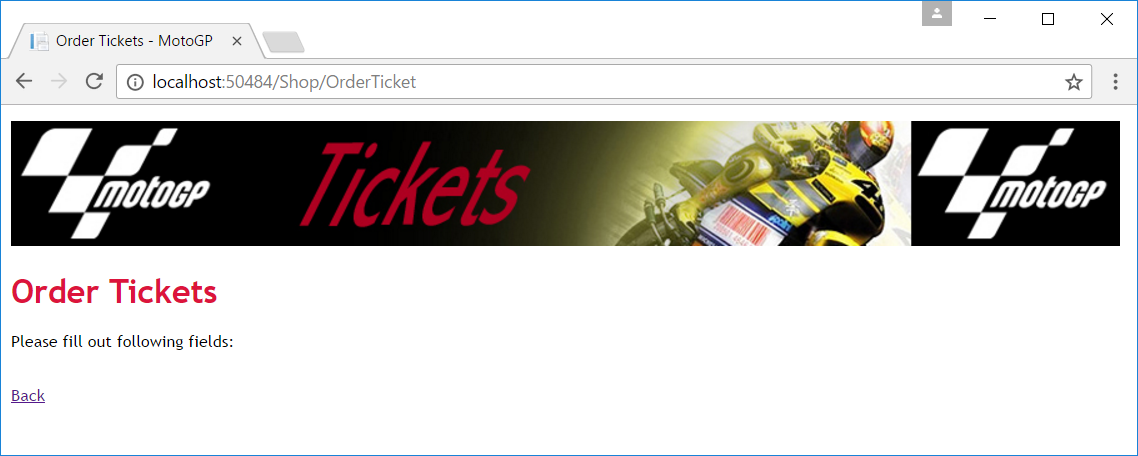
***MotoGP – part 4***

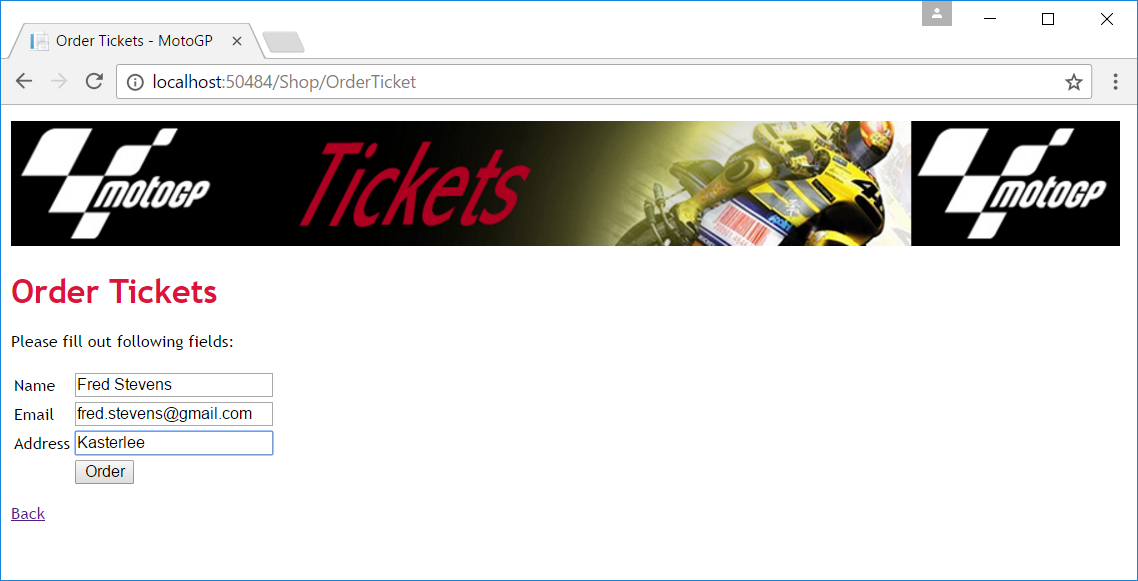
In part 4 of the exercise, we make sure that tickets can be ordered for our *MotoGP* races.

***1. Order Tickets***

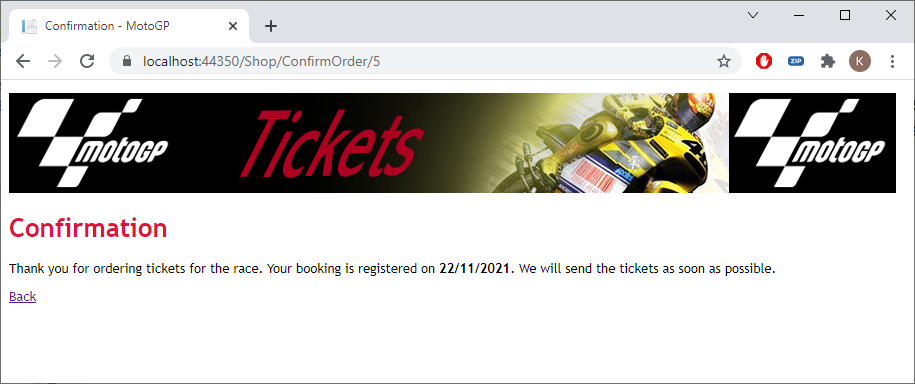
In the *Shop* controller, an *OrderTicket* method already exists that calls up the following view:



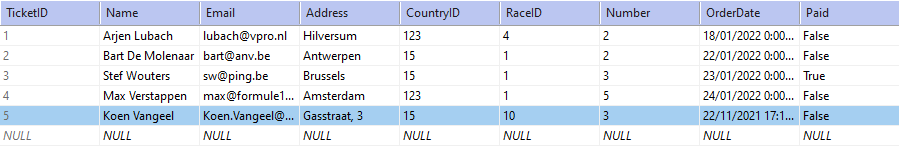
Adjust this view so that the person, who wants to order tickets, can fill in his/her details:



Now create a *Post* method so that the data is actually written to the database. For the *CountryID*, *RaceID* and *Number* fields, write the value 1 for the time being. *OrderDate* is set to the system date. The value of *Paid* is false. At the end do a redirect to the confirmation page:

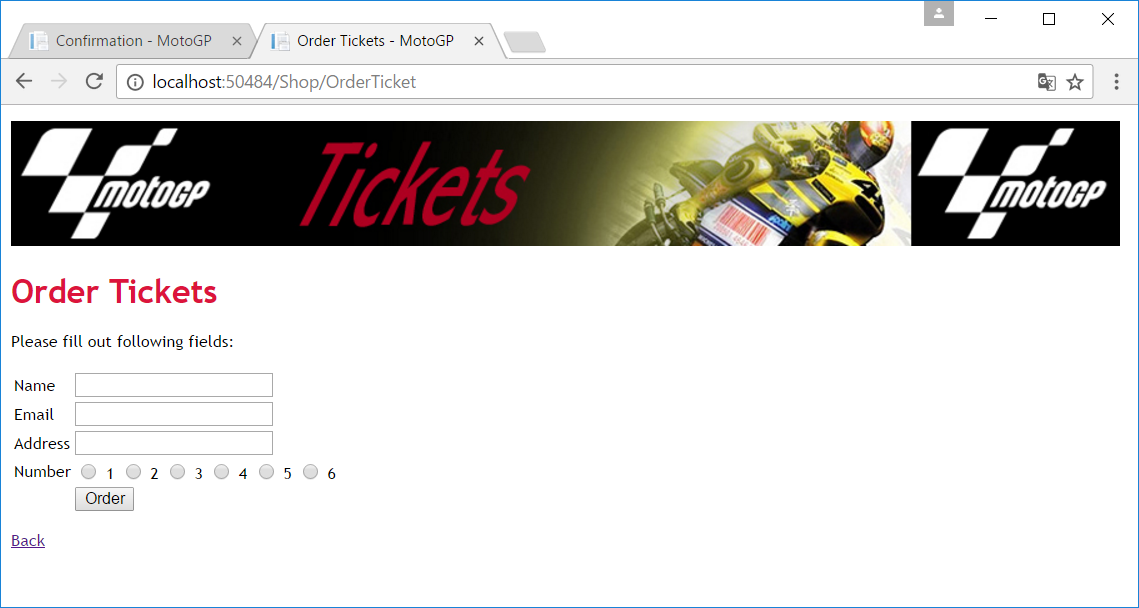


Please check if the data is saved correctly into the database table:



***2. Select the number of tickets***

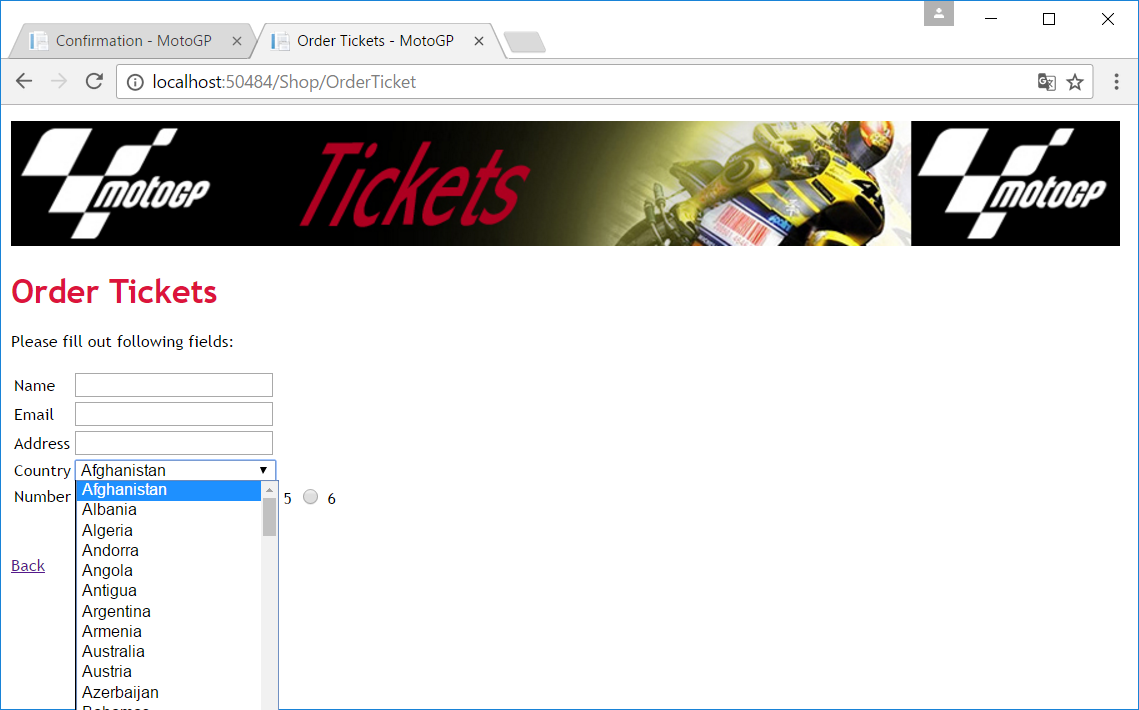
Adjust the view so that the number of tickets can also be selected. Provide a maximum number of six tickets per order:



Write the selected number into the database.

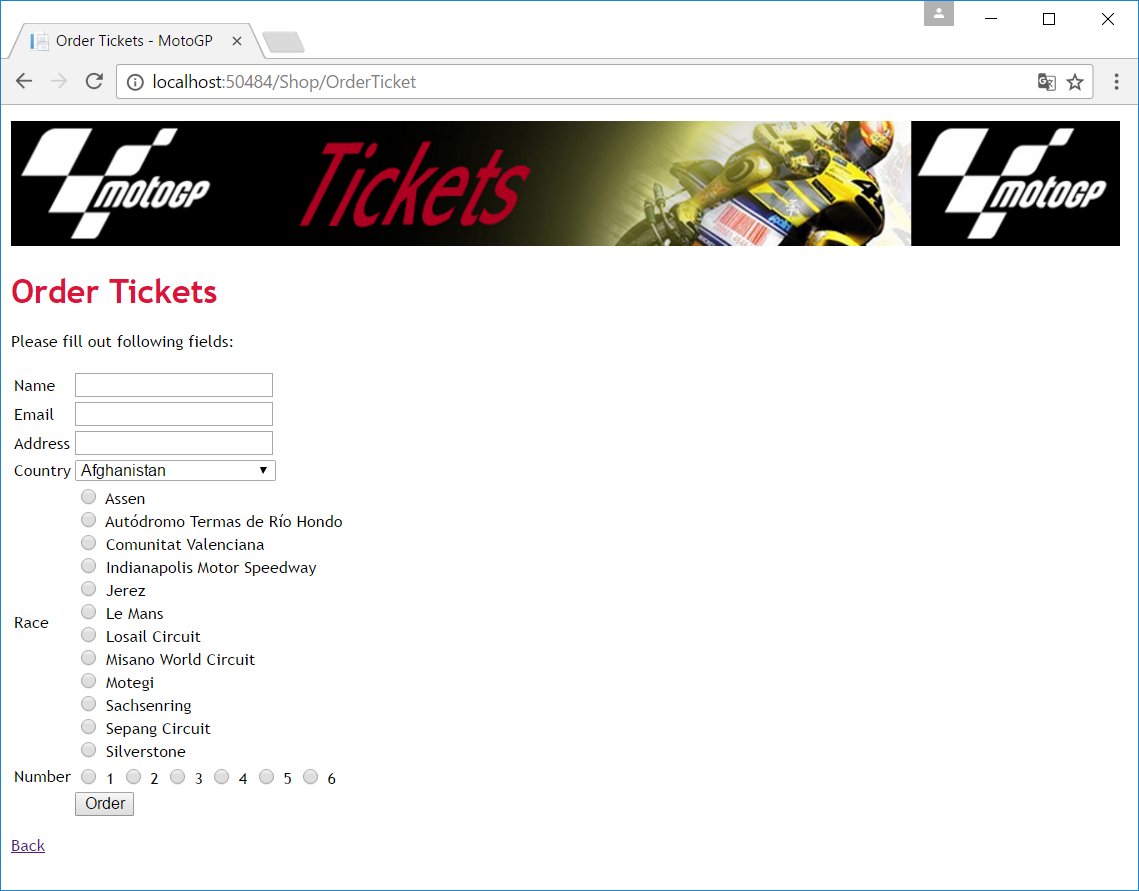
***3. Country***

Modify the controller and create a *SelectList* of all countries (**use the *Country* model**). Put a *select* on the view so that the users can select their country. The countries are shown alphabetically. Make sure the selected country is also written to the database.:



***4. Race***

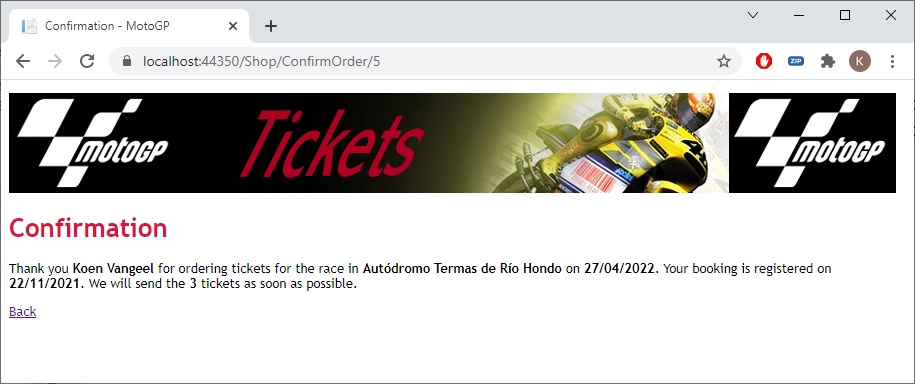
Generate a radio button for each race. The race info comes from the database of course:



And you also save the chosen race into the *Ticket* table.

***5. Confirmation***

Show a modified confirmation after the order has been placed:



Tip: You can pass a parameter *id* when redirecting to another page as follows:

return RedirectToAction("ConfirmOrder", new { id = 6 });

You must of course replace the number 6 with the *id* of the inserted ticket.

***6. Testing***

The tickets menu item should also be fully functional now. Please test via the menu:

