# Exercises: Introduction to ASP.NET Core

Problems for exercises and homework for the [“C# MVC Frameworks - ASP.NET Core” course @ SoftUni](https://softuni.bg/trainings/2197/csharp-mvc-frameworks-asp-net-core-november-2018).

# Eventures

**Eventures** **Inc**. is a fast-rising newly made Start-Up Company, which specializes in **Event Tickets Sales**. It is said to be the killer of systems like Eventim, Eventbride, etc.

You have been appointed as the developer of the **main web application**. This is a great responsibility, so do your best and do not dissapoint your employers. The application functionality is not that complex, and it will be **separated** into **several parts**, each part consisting of **several tasks**.

Your current task is to extend the functionality of the application, by adding Orders, and personal Events.

## Data Storage

Create the following data model:

### Order

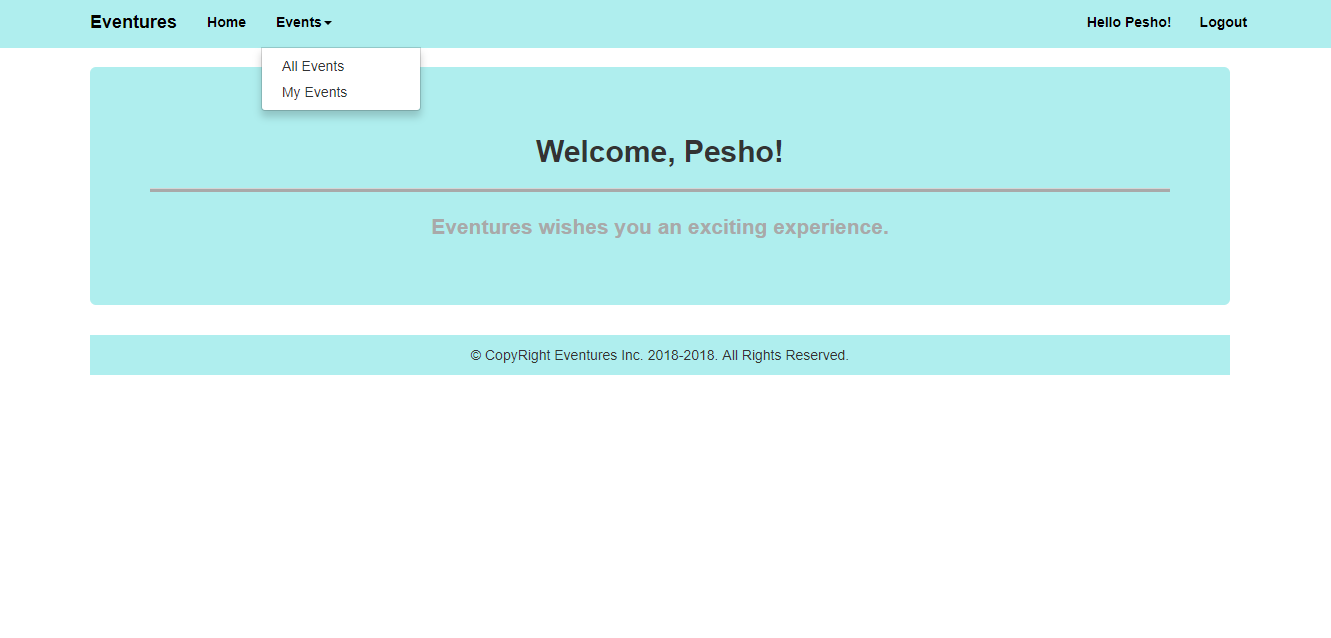
Has the following properties:

* Ordered On – a DateTime object.
* Event – an Event object.
* Customer – a User object.
* Tickets Count – an integer.

Each of the **data models**, also has an **Id**, which should be a **GUID**. Make the relationship between data models.

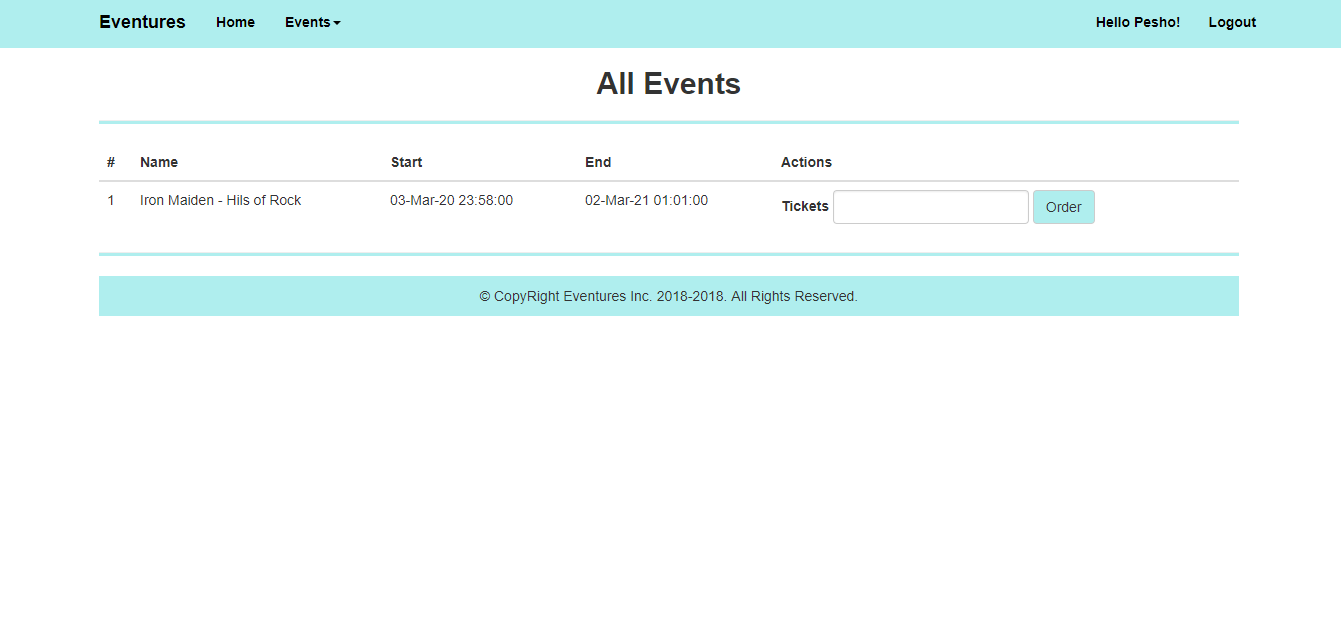
## Front-End

### Home (logged-in users)

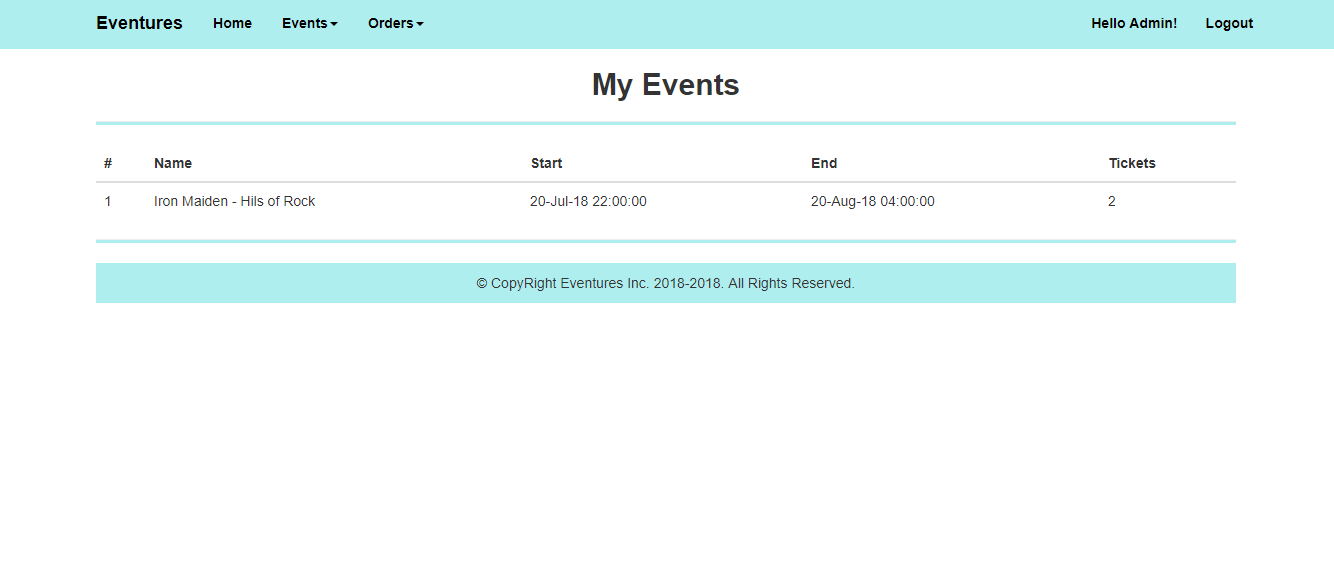


**Note**: Notice how we have [My Events] next to [All Events].

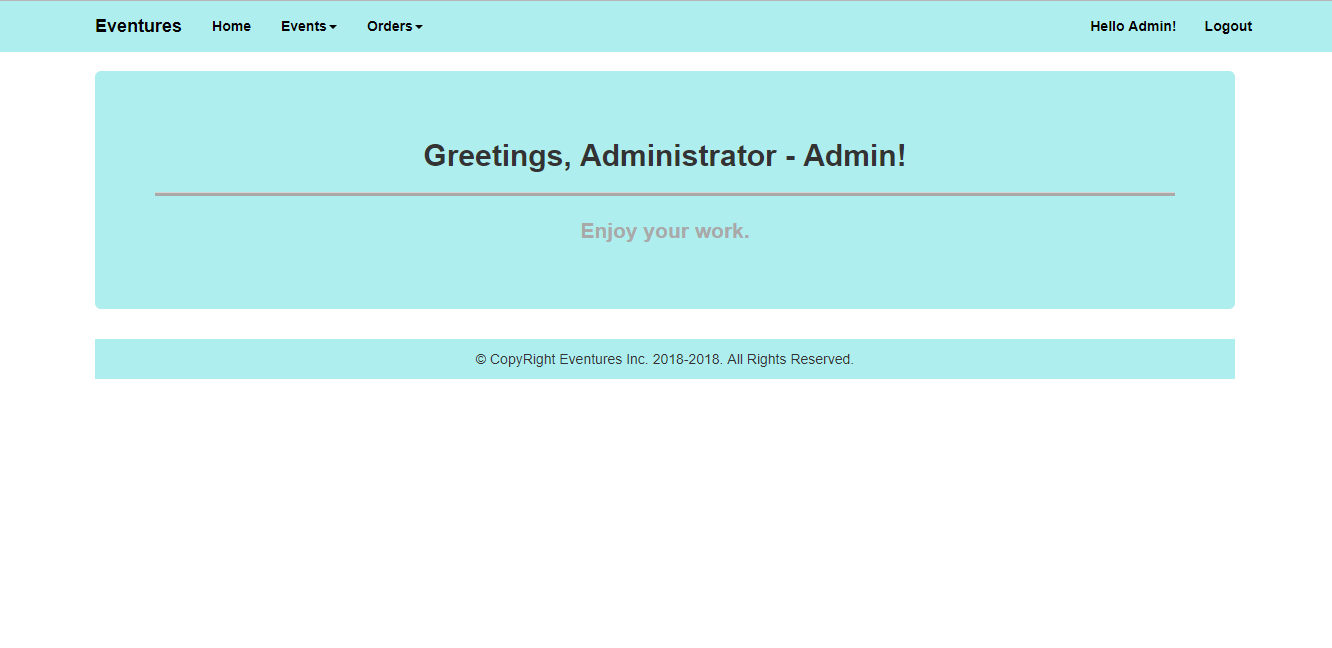
### All Events (logged-in)

**Note**: There is a simple **Form** for **Tickets** in the Actions tab.

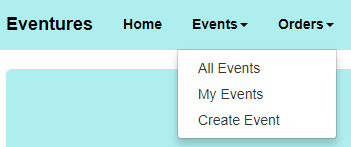
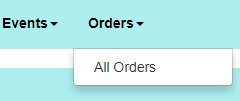
### My Events (logged-in)



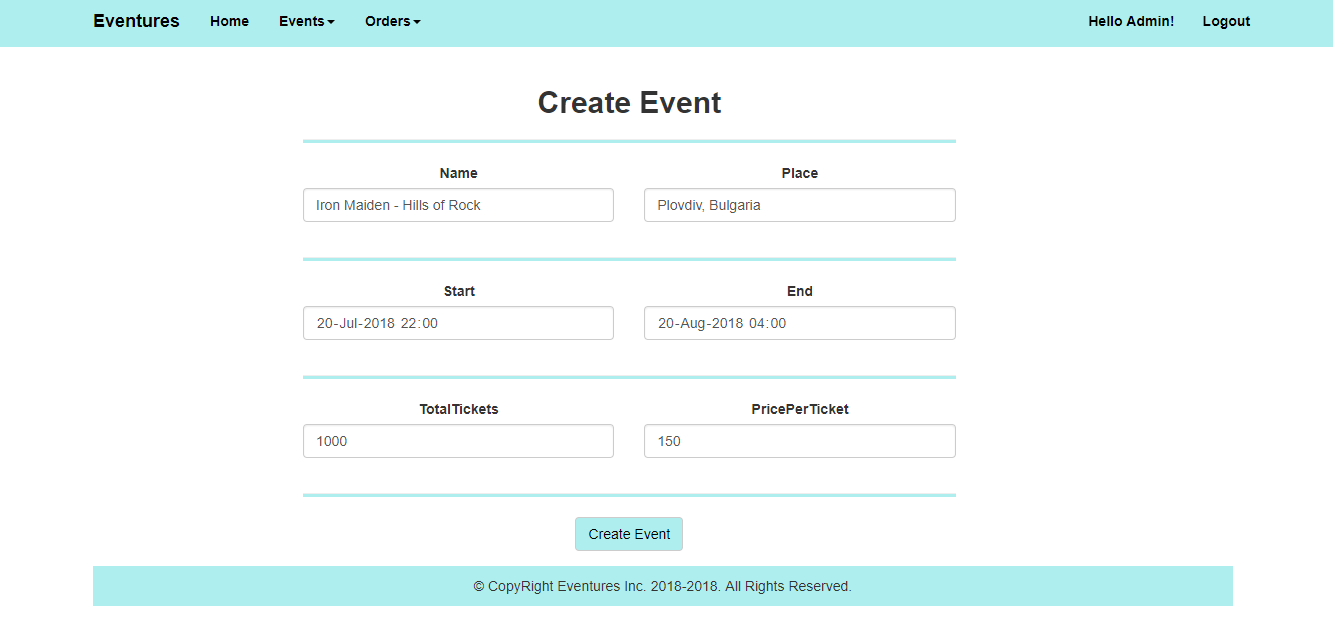
### Admin Home (logged-in Admin)



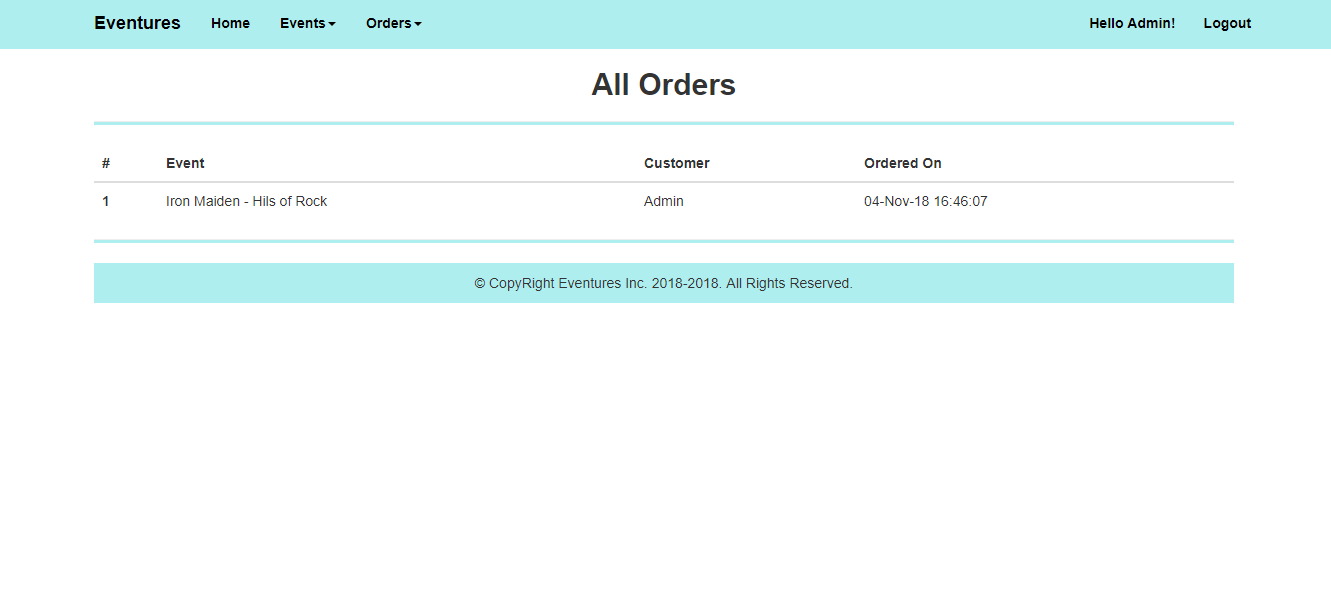
#### Admin Navigation



### Create Events



### All Orders



**Note**: The color for the application is #AFEEEE.

## Business Logic

### Technical Requirements

### Functionality

The application should provide its Guest users (**not logged-in**) the functionality to register and login.

The application should provide its Regular users (**logged-in** Users with Role – User) the functionality to **view all** Events, **order** **tickets** for them, and **view** all Events, **they’ve** ordered **tickets** for (**My Events**).

The application should provide its Admin users (**logged-in** Users with Role – Admin) the functionality to **create** **new** Events, **view all** Events, **order** **tickets** for them, **view** all Events, **they’ve** ordered **tickets** for (**My Events**), and **view** all Orders made.

Upon **clicking** the[Order] button, you should add new Order to the database.

* Each event has its own Tickets Order Form.
* Each submitted Tickets Order (Clicket [Order] button) should create an Order with the given Tickets, and the corresponding Event.