In this video we will discuss, customizing the auto-generated create view. Please [watch Part 26](http://csharp-video-tutorials.blogspot.com/2013/05/part-26-customizing-autogenerated-index.html), before proceeding.   
  
At the moment, none of the fields on **"Create"** view are required. This means,when you click on the **"Create"** button without filling any data, **NULL values** are stored in all the columns of **tblEmployee** table.  
  
**So, how to make these fields on the "Create" view required?**  
Add [Required] attribute to the **"Employee"** class. The **"Employee"** class that is present in **"EmployeeDataModel.Designer.cs"** is auto-generated by the **entity framework**. There is no point in adding the [Required] attribute to this class, as we will loose the changes if the class is auto-generated again.

To achieve this, add a class file with **"name=Employee.cs"** to **"Models"** folder.  
  
**Copy and paste the following code in "Employee.cs" file**  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
using System.ComponentModel.DataAnnotations;  
  
namespace MVCDemo.Models  
{  
    [MetadataType(typeof(EmployeeMetaData))]  
    public partial class Employee  
    {  
    }  
  
    public class EmployeeMetaData  
    {  
        [Required]  
        public string Name { get; set; }  
  
        [Required]  
        public string Gender { get; set; }  
  
        [Required]  
        public string City { get; set; }  
  
        [Required]  
        [Display(Name="Department")]  
        public int DepartmentId { get; set; }  
    }  
}  
  
At this point, run the application and click on the **"Create"** button without filling any data. Notice that we get validation error messages as expected. In a later video session, we will discuss changing the colour of the validation messages.  
  
If you want **"Select Department"** as the first item in the **"Department"** dropdownlist on **"Create"** view, then,   
**REPLACE THE FOLLOWING LINE**  
@Html.DropDownList("DepartmentId", String.Empty)  
  
**WITH**  
@Html.DropDownList("DepartmentId", "Select Department")  
  
Notice that, a textbox is used for gender. It is ideal to have a dropdownlist for gender rather than a textbox. To achieve this, make the following changes to **"Create.cshtml"**view.  
  
**REPLACE THE FOLLOWING CODE**  
@Html.EditorFor(model => model.Gender)  
  
**WITH**  
@Html.DropDownList("Gender", new List<SelectListItem>  
{  
new SelectListItem { Text = "Male", Value="Male" },  
new SelectListItem { Text = "Female", Value="Female" }  
}, "Select Gender")