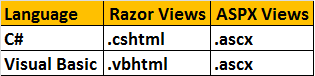
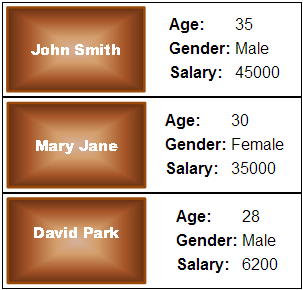
In this video we will discuss partial views in mvc.

If you are an asp.net web-forms developer, then you will realize that **partial views in mvc are similar to user controls in asp.net webforms.**   
  
Partial views are used to **encapsulate re-usable view logic** and are a great means to simplify the complexity of views. These partial views can then be used on multiple views, where we need similar view logic.

If you are using web forms view engine, then the partial views have the extension of **.ascx**. If the view engine is razor and programming language is c#, then partial views have the extension of **.cshtml**. On the other hand if the programming language is visual basic, then the extension is .vbhtml.  
   
  
Let us understand **partial views with an example**. We want to display, employee photo and his details as shown in the image below.  
   
  
**Index Action() method in HomeController retrurns the list of employees.**  
public ActionResult Index()  
{  
    SampleDBContext db = new SampleDBContext();  
    return View(db.Employees.ToList());  
}  
  
We will have the following code in Index.cshtml. This view is a bit messy and complex to understand.  
@model IEnumerable<MVCDemo.Models.Employee>  
@foreach (var item in Model)  
{  
    <table style="font-family:Arial; border:1px solid black; width: 300px">  
        <tr>  
            <td>  
                <img src="@Url.Content(item.Photo)" alt="@item.AlternateText" />  
            </td>  
            <td>  
                <table>  
                    <tr>  
                        <td><b>Age:</b></td>  
                        <td>@item.Age</td>  
                    </tr>  
                    <tr>  
                        <td><b>Gender:</b></td>  
                        <td>@item.Gender</td>  
                    </tr>  
                    <tr>  
                        <td><b>Salary:</b></td>  
                        <td>@item.Salary</td>  
                    </tr>  
                </table>  
            </td>  
        </tr>  
    </table>  
}  
  
To **simplify this view**, let's encapsulate the HTML and code that produces the employee table in a partial view.   
  
Right click on the **"Shared"** folder and add a view. Set  
View name = \_Employee  
View engine = Razor  
Create a strongly typed view = Checked  
Model class = Employee (MVCDemo.Models)  
Scaffold template = Empty  
Create as a partial view = Checked  
  
This should add **"\_Employee.cshtml"** partial view to the **"Shared"** folder.   
  
**Please note that**, partial views can be added to **"Shared"** folder or to a **specific views folder**. Partial views that are in the "Shared" folder are available for all the views in the entire project, where as partial views in a specific folder are available only for the views with-in that folder.

Copy and paste the following code in **"\_Employee.cshtml"** partial view  
@model MVCDemo.Models.Employee  
<table style="font-family:Arial; border:1px solid black; width: 300px">  
        <tr>  
            <td>  
                <img src="@Url.Content(Model.Photo)" alt="@Model.AlternateText" />  
            </td>  
            <td>  
                <table>  
                    <tr>  
                        <td><b>Age:</b></td>  
                        <td>@Model.Age</td>  
                    </tr>  
                    <tr>  
                        <td><b>Gender:</b></td>  
                        <td>@Model.Gender</td>  
                    </tr>  
                    <tr>  
                        <td><b>Salary:</b></td>  
                        <td>@Model.Salary</td>  
                    </tr>  
                </table>  
            </td>  
        </tr>  
</table>  
  
Now, make the following changes to **Index.cshtml** view. Notice that the view is much simplified now. To render the partial view, we are using **Partial**() html helper method. There are several overloaded versions of this method. We are using a version that expects **2 parameters**, i.e the name of the partial view and the model object.  
@model IEnumerable<MVCDemo.Models.Employee>  
@foreach (var item in Model)  
{  
    @Html.Partial("\_Employee", item)  
}