Passing data to view

3 ways to pass data to the View :
 ViewData
 ViewBag
Strongly Type View

ViewData and ViewBag resolved dynamically at runtime do not provide compile type checking
 and intallisense but strongly type does

ViewData

```
ViewData["PageTitle"] = "MyIndex";
<h1>@ViewData["PageTitle"]</h1>
```

any type rather than string need to be casted to type

```
UserIdentityUser user = new UserIdentityUser();
user.Description = "My Desc";
ViewData["User"] = user;
//////////////////////////// In the view //////////////////
@using dotNetCore_Identity.Models;
@{
    var user = @ViewData["User"] as UserIdentityUser;
    var desc=user.Description;
}
    <h1>@desc</h1>
```

ViewBag

```
ViewBag.Title = "My Some Title";
<h1>@ViewBag.Title</h1>
```

Strongly Type View

```
  UserIdentityUser user = new UserIdentityUser();
  user.Description = "My Desc";
return View(user);
//////////////////// in the view ///////////////////////////
@using dotNetCore_Identity.Models;
@model UserIdentityUser
<h1>@Model.Description</h1>
```

viewModels are Strongly Type View like HomeDetailsViewModel (ControllerActionViewModel)

```
  @RenderSection("SectionName", false) <!--  bool value is it required  // default e true
-->
```

a section in the layout view is rendered where `@RenderSection() is called`
od viewata koi se renderiraat ako imaat sekcija definirano ke se renderiraat kaj `@RenderSection()`

```
@section SectionName{
 <h1>SectionName</h1>
}
```

Useful Razor Content Expressions

Razor is smart enough to know that the **space** character after the expression is not a valid identifi er, so it transitions smoothly back into markup

@: Out of Stock ... The @: characters prevent Razor from interpreting this as a C# statement, which is the default behavior when it encounters text

ViewStart ViewImports Layout

_VIewStart _ViewImports _Layout are hierarchical can be overriden

Layout

It is important to understand the difference between omitting the Layout property from the view file and setting it to null. If your view is self-contained and you do not want to use a layout, then set the Layout property to null. If you omit the Layout property, then MVC will assume that you do want a layout and that it should use the value it finds in the view start file