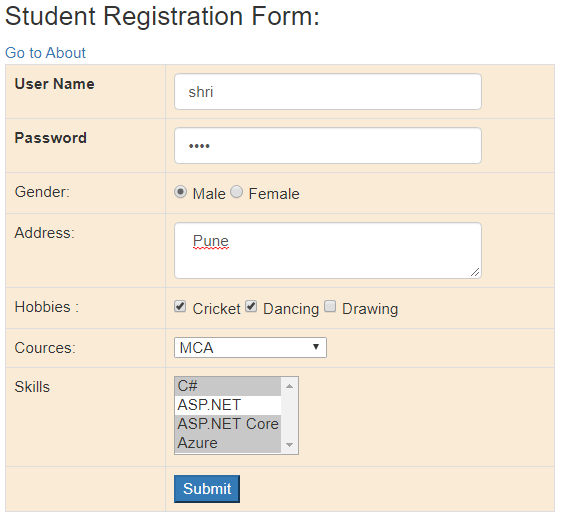
We will also design a student registration page, shown below, using standard HTML helpers.



**In this article, you will learn the following points about Standard HTML Helpers in MVC 5.**

* What is a Standard HTML helper in Asp.NET MVC5?
* List of standard HTML helpers
* How to use Standard HTML helpers on View?.
* What are some examples of Standard HTML helpers?
* Advantages of using Standard HTML Helpers
* Interview Question & Home Work

**The following are the previous top articles on ASP.NET MVC 5**

* [HTML Helpers with example In MVC 5](https://www.c-sharpcorner.com/article/html-helpers-in-asp-net-mvc-5/)
* [Razor View Engine In ASP.NET MVC 5](https://www.c-sharpcorner.com/article/razor-view-engine-in-asp-net-mvc-5/)
* [Model in ASP.NET MVC 5](https://www.c-sharpcorner.com/article/model-in-asp-net-mvc-5/)
* [Action Method in MVC 5 with example](https://www.programmingwithshri.com/2019/05/action-method-in-asp.net-mvc-5.html)

**What is a Standard HTML helper in Asp.NET MVC5?**

* Standard HTML helpers are used to render the most common type of HTML controls like Label,TextBox, Password, TextArea, CheckBox, RadioButtion, DropDownList, Listbox,Display,Editor and ActionLink etc.
* HTML helpers always start with @HTML. Ther are an object of Html helper class.@ symbol used to access the server-side code. The extension method of the HTML helper class has several overloaded methods. We can use it as per our requirement.
* HTML is a property of type HtmlHelper included in the base class of razor view WebViewPage. TextBox() or TextBoxFor()... are extension methods included in HtmlHelper class.
* The HtmlHelper class generates HTML elements. For example,@Html.ActionLink("Add Empoyee", "Create","Employee") would generate the anchor tag <a href="/Employee/Create">Add Empolyee</a>.

**List of Standard HTML Helpers In ASP.NET MVC 5**

* @Html.TextBox
* @Html.Password
* @Html.TextArea
* @Html.CheckBox
* @Html.RadioButton
* @Html.DropDownList
* @Html.ListBox
* @Html.Hidden
* @Html.Display
* @Html.Editor
* @Html.ActionLink
* @Html.BeginForm
* @Html.Label

**@Html.TextBox()**

TextBox() helper method is a loosely typed extension method that is used to create textbox(<input type=”text”>) element in razor view.

@Html.TextBox() method inside the “System.Web.Mvc.Html” namespace. The return type is “MvcHtmlString”.

**TextBox() Method Signature**

1. MvcHtmlString Html.TextBox(string name, string value, object htmlAttributes)

The TextBox() method is a loosely typed method because it has a name parameter that is a string. It automatically displays the value of the model property in a textbox and vice-versa. The name parameter can be a model class property. The value should be an object value of the property. The third parameter is that the HTML attribute can provide CSS or Id to the textbox. It has several overload methods we can use as per our requirement.

**How to use TextBox() in Razor view**

1. @Html.TextBox("txtUserName", "", **new** { @**class** = "form-control" })

**Html Result**

1. <input **class**="form-control" id="txtUserName" name="txtUserName" type="text" value="">

**Output**



**How to set default value to TextBox() in Razor view**

1. @Html.TextBox("txtUserName", "Test Value", **new** { @**class** = "form-control" })

**@Html.Password()**

The HTML helper class is used to generate HTML elements. The Password() helper method is a loosely typed extension method that is used to create textbox(<input type=”password”>) element in razor view.

@Html.Password() method inside the “System.Web.Mvc.Html” namespace. The return type is “MvcHtmlString”.

**Password() Method Signature**

1. MvcHtmlString Html.Password(string name, string value, object htmlAttributes)

The Password() method is a loosely typed method because its name parameter is a string. The name parameter can be a model class property. The value should be an object value of the property. So it automatically displays a value of the model property in a password and visa-versa. The third parameter is that the HTML attribute can provide CSS or Id to the Password. It has several overload methods we can use as per our requirement.

**How to use Password() Helper in Razor view**

1. @Html.Password("Password", "", **new** { @**class** = "form-control" }

**Html Result**

1. <input **class**="form-control" id="Password" name="Password" type="password" value="">

**Output**



**@Html.TextArea()**

TextArea() helper method is loosely typed extension method used to create a TextArea(<textarea cols="20" id="Address" name="Address" rows="2"> </textarea>) element in razor view.

@Html.TextArea() method inside the “System.Web.Mvc.Html” namespace. The return type is “MvcHtmlString”.

**TextArea() Method Signature**

1. MvcHtmlString Html.TextArea(string name, string value, object htmlAttributes)

The TextArea() method is a loosely typed method because its name parameter is a string. The name parameter can be model class property. The value should be an object value of the property. It automatically displays a value of the model property in a textarea and visa-versa. The third parameter is that an HTML attribute can provide CSS, Id to the TextArea. It has several overload methods we can use as per our requirement.

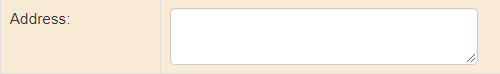
**How to use TextArea() Helper in Razor view**

1. @Html.TextArea("Address", " ", **new** { @**class** = "form-control",id="IdAddress" })

**Html Result**

1. <textarea **class**="form-control"
2. cols="20"
3. id="IdAddress"
4. name="Address"
5. rows="2">
6. </textarea>

**Output**



**@Html.CheckBox():**

This HTML helper class is used to generate HTML elements. The CheckBox() helper method is a loosely typed extension method which is used to create CheckBox(<input type="checkbox" >) element in razor view.

@Html.CheckBox() method inside the “System.Web.Mvc.Html” namespace. The return type is “MvcHtmlString”.

**CheckBox() Method Signature**

1. MvcHtmlString CheckBox(string name, bool isChecked, object htmlAttributes)

The CheckBox() method is a loosely typed method because its name parameter is a string. The name parameter can be model class property. The value should be an object value of the property, so it automatically displays a value of the model property in a CheckBox and visa-versa. The third parameter is an HTML attribute that can add CSS and the Id to the CheckBox. It has several overload methods we can use as per our requirement.

**How to use CheckBox() Helper in Razor view**

1. @Html.CheckBox("Dancing")

**How to set CheckBox() default as checked**

1. @Html.CheckBox("Cricket", **true**)

**Html Result**

1. <input checked="checked" id="Cricket" name="Cricket" type="checkbox" value="true">
2. <input id="Dancing" name="Dancing" type="checkbox" value="false">

**Output**



**@Html.RadioButton()**

This HTML helper class is used to generate HTML elements. The RadioButton() helper method is a loosely typed extension method which is used to create RadioButton(<input type="radio" >) element in razor view.

@Html.RadioButton() method inside the “System.Web.Mvc.Html” namespace. And return type is “MvcHtmlString”.

**RadioButton() Method Signature**

1. MvcHtmlString RadioButton(string name, object value, bool isChecked, object htmlAttributes)

The RadioButton() method is a loosely typed method because its name parameter is a string. The name parameter can be model class property. The value should be an object value of the property so that it automatically displays a value of the model property in a RadioButton and visa-versa. The third parameter is an HTML attribute that can add CSS and Id to the RadioButton. It has several overload methods we can use as per our requirement.

**How to use RadioButton() Helper in Razor view**

**Example:**Male and Female Gender

1. @Html.RadioButton("Gender", "Male", **true**, **new** { id = "male" }) Male
2. @Html.RadioButton("Gender", "Female", **false**, **new** { id = "female" }) Female

**How to set RadioButton() default as checked**

1. @Html.RadioButton("Gender", "Male", **true**, **new** { id = "male" }) Male

**Html Result**

1. <input **checked**="checked" id="male" name="Gender" type="radio" value="Male">
2. <input id="female" name="Gender" type="radio" value="Female">

**Output**



**@Html.DropDownList()**

The DropDownList() helper method is a loosely typed extension method used to create a DropDownList(<select id="" name="">) element in razor view with a specified name, list items and HTML attributes.

@Html.DropDownList() method inside the “System.Web.Mvc.Html” namespace. The return type is “MvcHtmlString”.

**DropDownList() Method Signature**

1. MvcHtmlString Html.DropDownList(string name, IEnumerable<SelectLestItem> selectList, string optionLabel, object htmlAttributes)

The DropDownList() method is a loosely typed method because it has a name parameter that is a string. The name parameter can be a model class property. The second parameter included is the list. It automatically displays the value of the model property in a DropDownList and visa-Versa. The third parameter is an HTML attribute that can add CSS, Id to the DropDownList. It has several overload methods we can use as per our requirement.

**How to use DropDownList() Helper in Razor view**

1. @{
2. IEnumerable<**string**> strList = **new** List<**string**> { "BCA", "BCS", "MCA", "MCS" };
3. }
4. @Html.DropDownList("ddlCourse", **new** SelectList(strList, strList.FirstOrDefault()), "--Select Course----")

**Html Result**

1. <select id="ddlCourse" name="ddlCourse">
2. <option value="">--Select Course----</option>
3. <option selected="selected">BCA</option>
4. <option>BCS</option>
5. <option>MCA</option>
6. <option>MCS</option>
7. </select>

**Output**



**@Html.ListBox()**

This HTML helper class is used to generate HTML elements. ListBox() helper method is a loosely typed extension method which is used to create ListBox(<select id="" multiple="multiple" name="">) element in razor view with specified name, list items and HTML attributes.

@Html.ListBox() method inside the “System.Web.Mvc.Html” namespace. The return type is “MvcHtmlString”.

**ListBox() Method Signature**

1. MvcHtmlString Html.ListBox(string name, IEnumerable<SelectLestItem> selectList, string optionLabel, object htmlAttributes)

The ListBox() method is a loosely typed method because it has a string name parameter. The name parameter can be a model class property. The value should be an object value of the property. Therefore, it automatically displays a value of the model property in a ListBox and visa-versa. The third parameter is an HTML attribute that can add CSS, Id to the ListBox. It has several overload methods we can use as per our requirement.

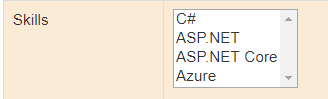
**How to use ListBox() Helper in Razor view**

1. @Html.ListBox("Select Skills",**new** List<SelectListItem> {
2. **new** SelectListItem{Text= "C#",Value="1"},
3. **new** SelectListItem{ Text="ASP.NET",Value="2" },
4. **new** SelectListItem{ Text="ASP.NET Core",Value="3" },
5. **new** SelectListItem{ Text="Azure",Value="4" }
6. })

**Html Result**

1. <select id="Select\_Skills" multiple="multiple" name="Select Skills">
2. <option value="1">C#</option>
3. <option value="2">ASP.NET</option>
4. <option value="3">ASP.NET Core</option>
5. <option value="4">Azure</option>
6. </select>

**Output**



**@Html.Label()**

This HTML helper class is used to generate HTML elements. The Label() helper method is a loosely typed extension method used to create Label(<label>) element in razor view.

@Html.Label() method inside the “System.Web.Mvc.Html” namespace. The return type is “MvcHtmlString”.

**Label() Method Signature**

1. MvcHtmlString Html.Label(string name, string value, object htmlAttributes)

The Label() method is a loosely typed method because its name parameter is a string. The name parameter can be a model class property. The value should be an object value of the property. Therefore, it automatically displays the value of the model property in a Label and visa-versa. The third parameter is an HTML attribute which can provide CSS and the Id to the Password. It has several overload methods that we can use as per our requirement.

**How to use Label() Helper in Razor view**

1. @Html.Label("User Name ")

**Html Result**

1. <label **for**="User\_Name\_">User Name </label>

**@Html.ActionLink()**

Html helper class is used to generate HTML elements. The ActionLink() helper method is loosely typed extension method which is used to create a Label(<a href="/"> element in razor view.

@Html.ActionLink() method inside the “System.Web.Mvc.Html” namespace. The return type is “MvcHtmlString”.

**ActionLink() Method Signature**

1. MvcHtmlString ActionLink(string linkText, string actionName, object routeValues, object htmlAttributes);

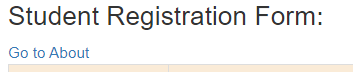
**How to use ActionLink() Helper in Razor view**

1. @Html.Label("User Name ")

**Html Result**

1. <a href="/Home/About">Go to About</a>

**Output**



**@Html.BeginForm():**

This HTML helper class is used to generate HTML elements. The BeginForm() helper method is loosely typed extension method which is used to create a Form (<form action="/" method="post">) element in razor view.

@Html.BeginForm() method is inside the “System.Web.Mvc.Html” namespace. The return type is “MvcHtmlString”.

**BeginForm() Method Signature**

1. MvcHtmlString ActionLink(string linkText, string actionName, object routeValues, object htmlAttributes);

**How to use BeginForm() Helper in Razor view**

1. @using (Html.BeginForm("Index", "Home", FormMethod.Post))
3. {
4. // Code here..
5. }

**Html Result**

1. <form action="/" method="post">
2. </form>

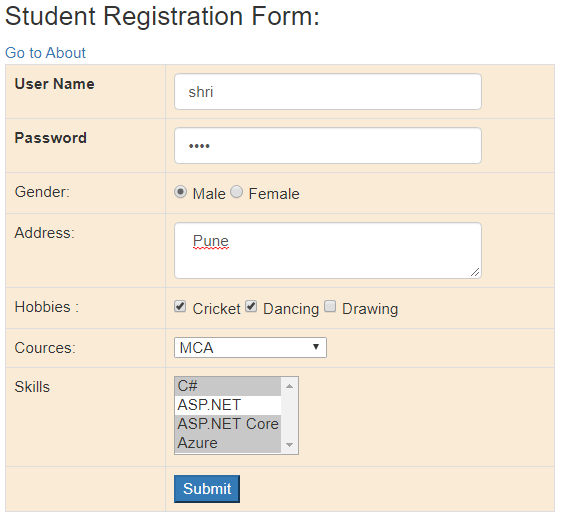
**How to use Standard HTML helpers on View?**

We will create a student registration form using Standard Html Helper in MVC 5.

Add the following code in index.cshtml page

1. <div class="container">
2. <h3>Student Registration Form: </h3>
3. @Html.ActionLink("Go to About", "About", new { })
4. @Html.Hidden("StudentId", "100")
5. @using (Html.BeginForm("Index", "Home", FormMethod.Post))
6. {
7. <div>
8. <table border="1" class="table table-bordered" style="background-color:antiquewhite;width:500px;">
9. <tbody>
10. <tr>
11. <td>
12. @Html.Label("User Name ")
13. </td>
14. <td>
15. @Html.TextBox("txtUserName", "", new { @class = "form-control" })
16. </td>
17. </tr>
18. <tr>
19. <td>
20. @Html.Label("Password", "Password")
21. </td>
22. <td>
23. @Html.Password("Password", "", new { @class = "form-control" })
24. </td>
25. </tr>
26. <tr>
27. <td>
28. Gender:
29. </td>
30. <td>
31. @Html.RadioButton("Gender", "Male", true, new { id = "male" }) Male
32. @Html.RadioButton("Gender", "Female", false, new { id = "female" }) Female
33. </td>
34. </tr>
35. <tr>
36. <td>
37. Address:
38. </td>
39. <td>
40. @Html.TextArea("Address", " ", new { @class = "form-control", id = "IdAddress" })
41. </td>
42. </tr>
43. <tr>
44. <td>
45. Hobbies :
46. </td>
47. <td>
48. @Html.CheckBox("Cricket", true) Cricket
49. @Html.CheckBox("Dancing") Dancing
50. @Html.CheckBox("Drawing") Drawing
52. </td>
53. </tr>
55. <tr>
56. <td>
57. Cources:
58. @{
59. IEnumerable<string> strList = new List<string> { "BCA", "BCS", "MCA", "MCS" };
60. }
61. </td>
62. <td>
63. @Html.DropDownList("ddlCourse", new SelectList(strList, strList.FirstOrDefault()), "--Select Course----")
64. </td>
65. </tr>
66. <tr>
67. <td>Skills</td>
68. <td>
69. @Html.ListBox("Select Skills",new List<SelectListItem> {
70. new SelectListItem{Text= "C#",Value="1"},
71. new SelectListItem{ Text="ASP.NET",Value="2" },
72. new SelectListItem{ Text="ASP.NET Core",Value="3" },
73. new SelectListItem{ Text="Azure",Value="4" }
74. })
75. </td>
77. </tr>
78. <tr>
79. <td>
81. </td>
82. <td>
83. <input type="submit" class="btn-primary" />
84. </td>
85. </tr>
86. </tbody>
87. </table>
88. </div>
89. }
90. </div>

**Output**



**Advantages of using Standard HTML Helpers**

* It is easy to use and simple to understand.
* They are loosely typed HTML helpers.