ecome a member C# Corner Login

Post Ask Question

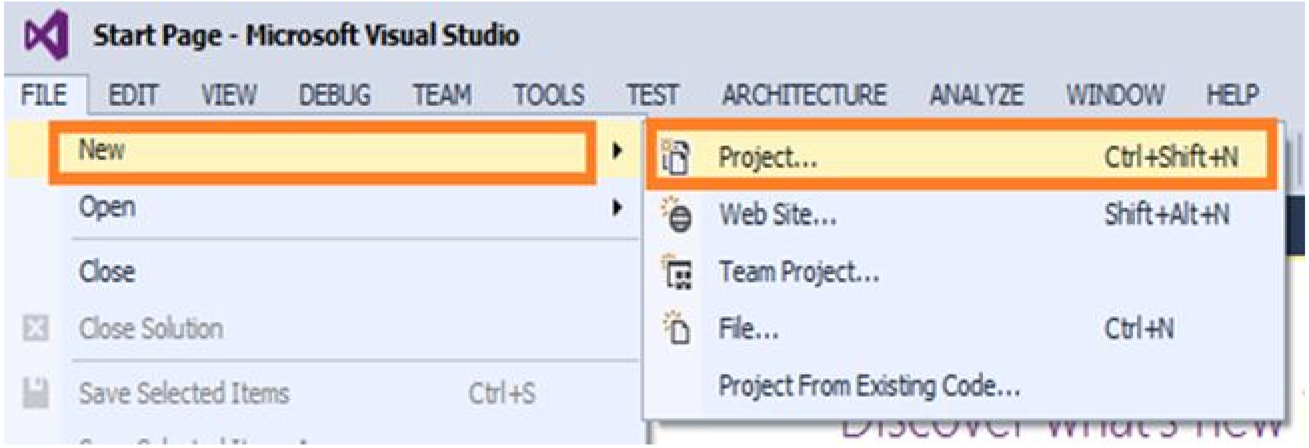
How To Create Multiple Languages In ASP.NET MVC 4.5 Framework

Avinash Thakur

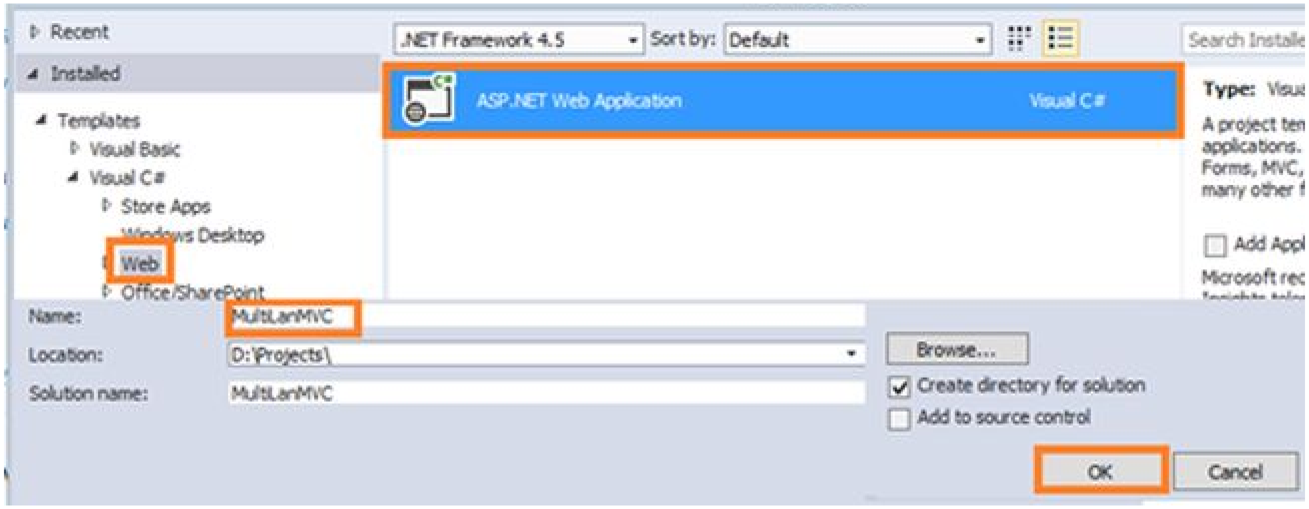
Updated date Dec 31, 2020

192.5k 39 15

Step 1 - Open your Visual Studio and click File=> New=>Project, as shown below-



Step 2 - Now, select ASP.NET Web Application template and give a name to your project. Here, our project name is “MultiLanMVC” and click OK button.



.NET Framework 4.5 Sort by: Default

Name: MultiLanMVC

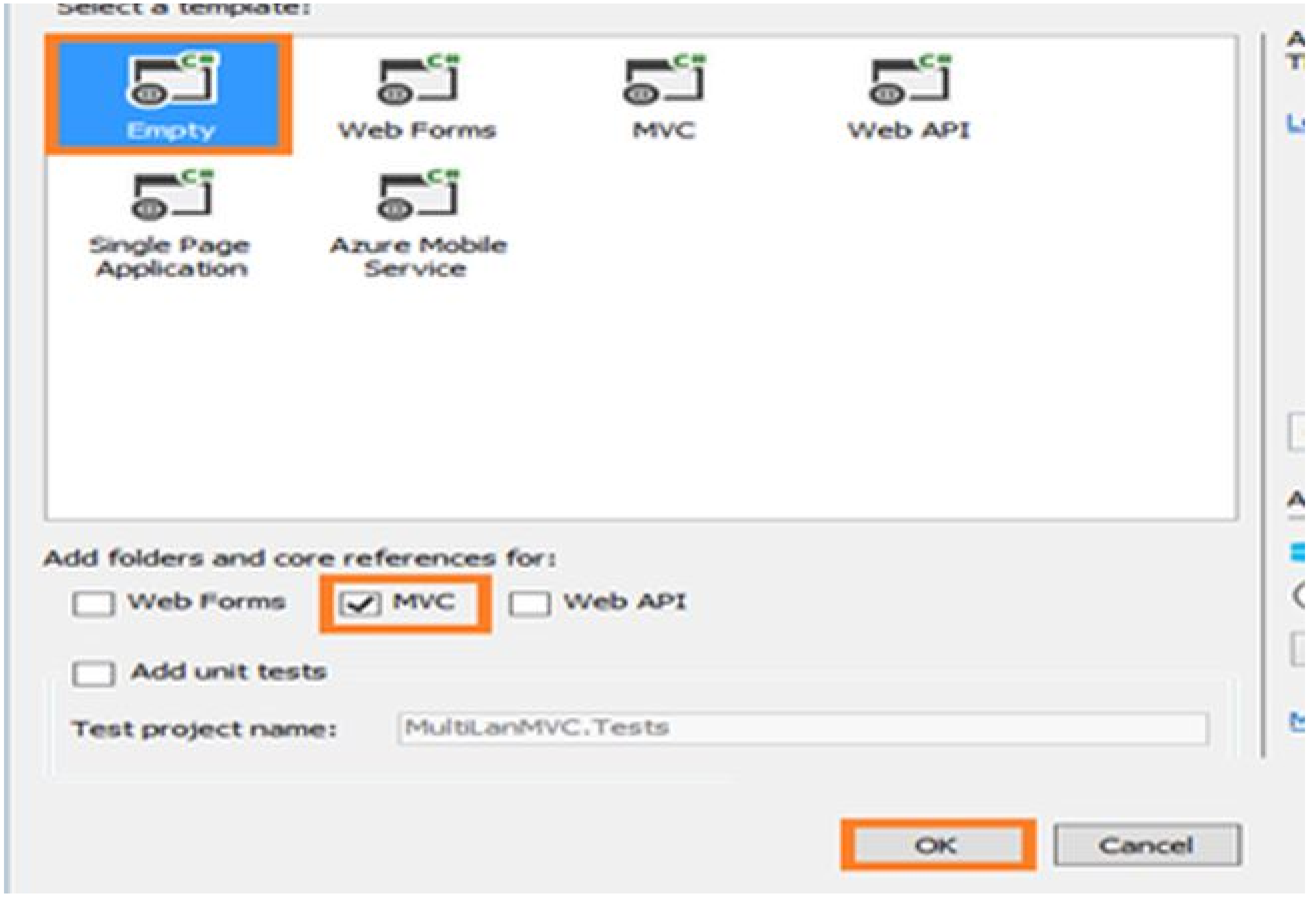
Location: D:\Projects\

Solution name: MultiLanMVC

☑ Create directory for solution

□ Add to source control

Step 3 - Select “Empty MVC template” and click OK button.



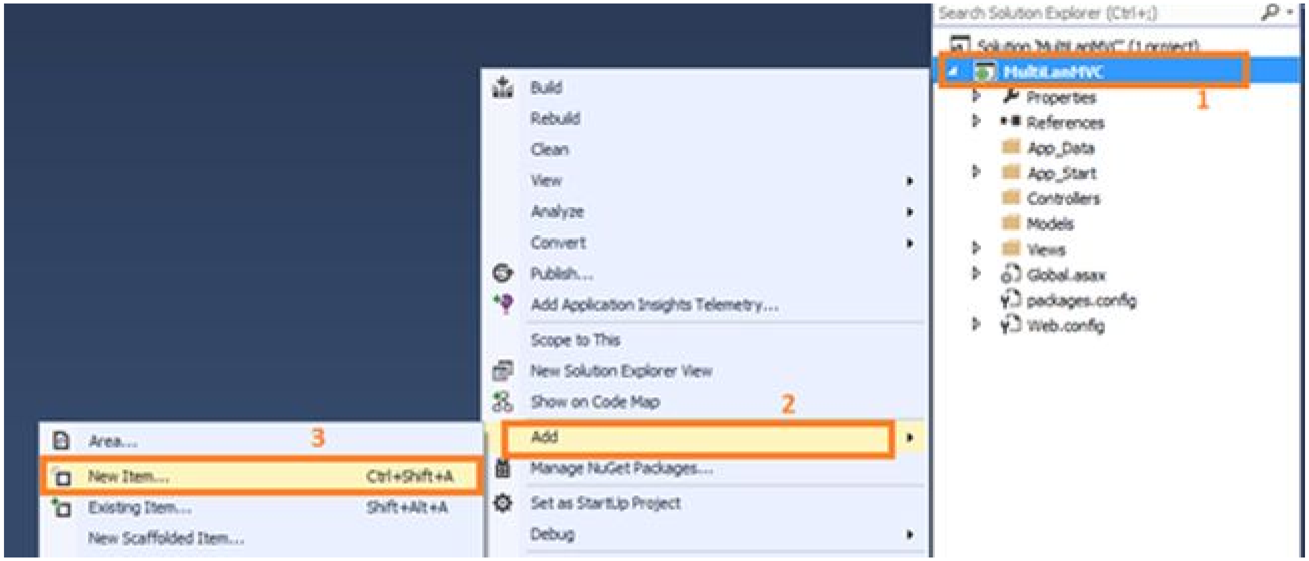
Add folders and core references for:

□ Web Forms ☑ MVC □ Web API

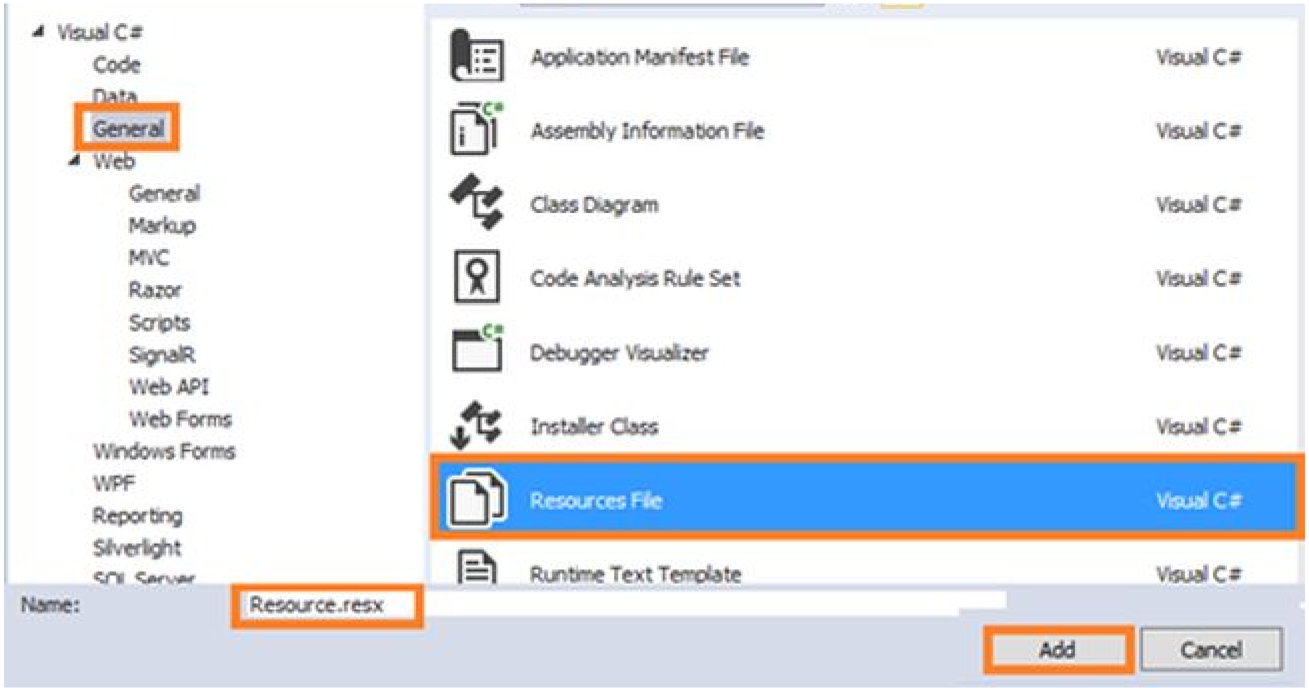
□ Add unit tests

Test project name: MultiLanMVC.Tests

Step 4 - You have to go to Solution Explorer and right click on your Project name=> Add=>New item.

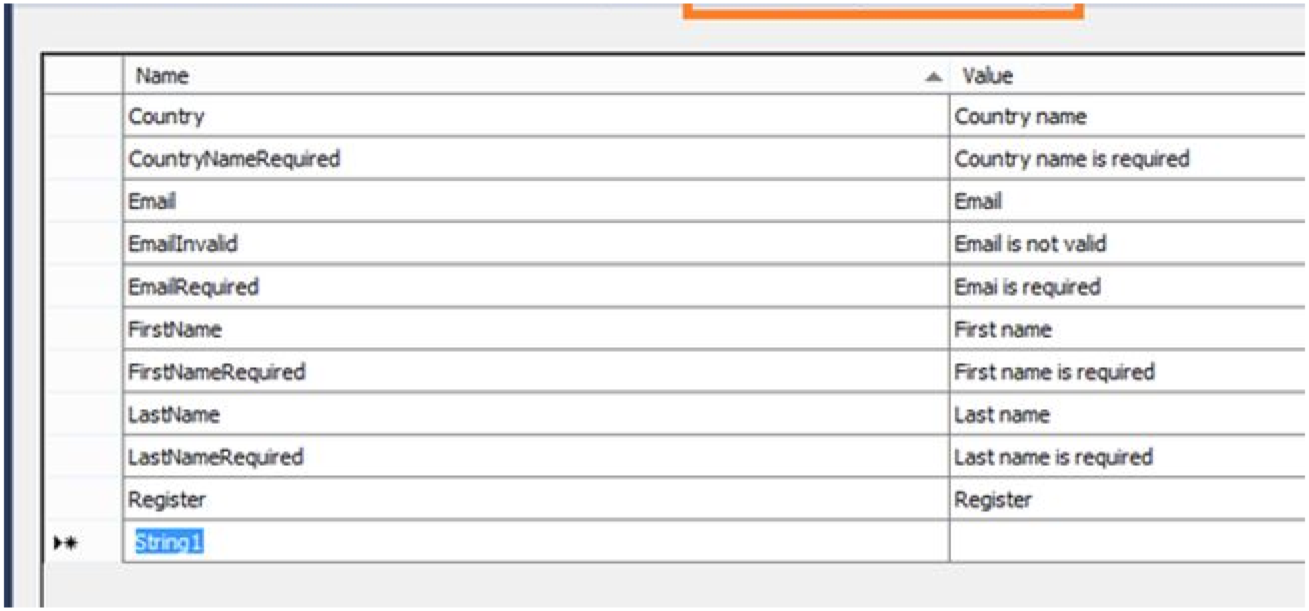


Step 5 - In this section, you have to add “Resources File”.


Visual C#
Code
Data
General
Web
General
Markup
MVC
Razor
Scripts
SignalR
Web API
Web Forms
Windows Forms
WPF
Reporting
Silverlight
SQL Server
Application Manifest File Visual C#
Assembly Information File Visual C#
Class Diagram Visual C#
Code Analysis Rule Set Visual C#
Debugger Visualizer Visual C#
Installer Class Visual C#
Resources File Visual C#
Runtime Text Template Visual C#
Add
Cancel

Name: Resource.resx

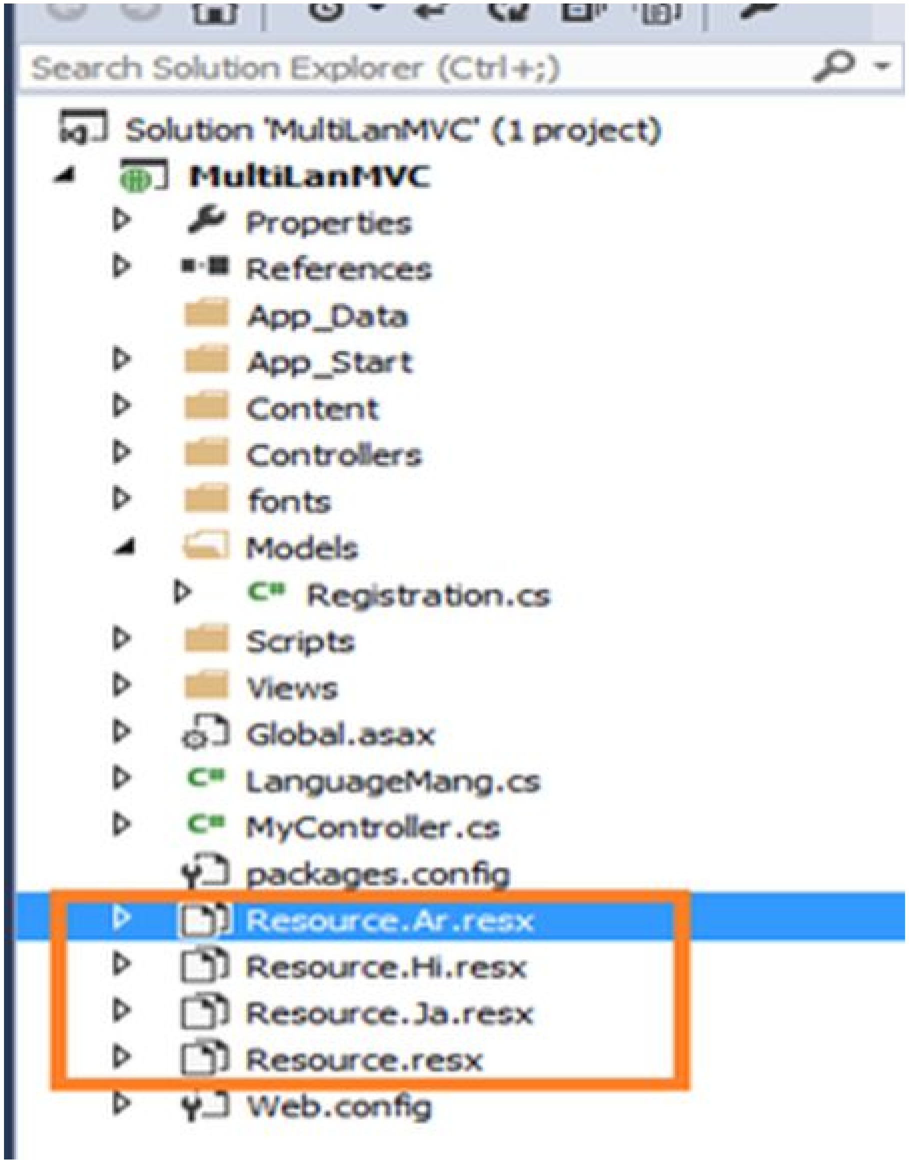
Step 6 - In this section, I have used Field name and its value declaration. You can click Access Modifier then select public access modifier. Don’t forget to change all the resource files Access Modifier to public.



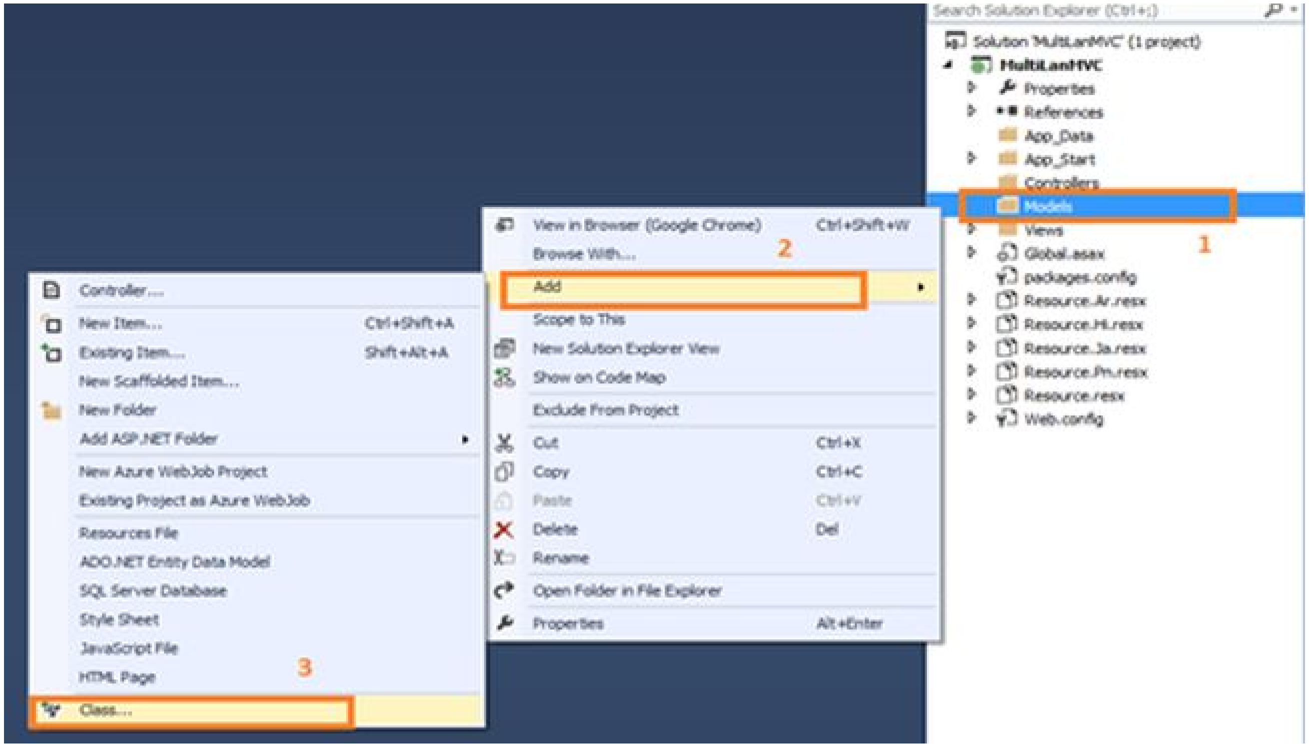
	Name	Value
	Country	Country name
	CountryNameRequired	Country name is required
	Email	Email
	EmailInvalid	Email is not valid
	EmailRequired	Emai is required
	FirstName	First name
	FirstNameRequired	First name is required
	LastName	Last name
	LastNameRequired	Last name is required
	Register	Register
▸*	String1	

Step 7 - Here, I have added four resource files for four different languages. I'm discussing here four resource files, which are-

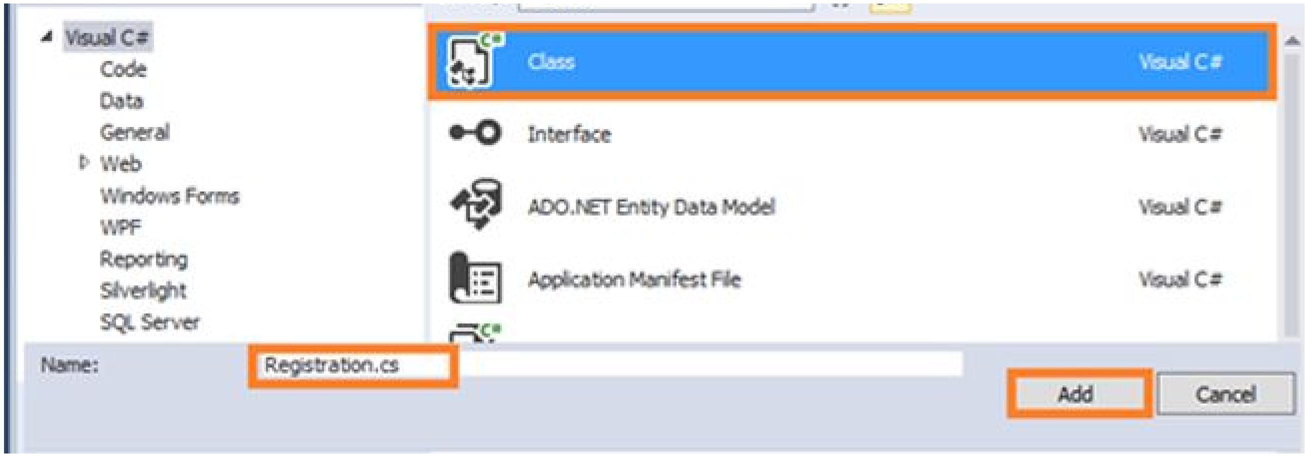
- *Resource.resx*- It is used for the default resource file associated with English. Here ".resx" is an extension file name in Resource file.
- *Resource.Hi.resx*- This resource file is used for Hindi language.
- *Resource.Ja.resx*- This resource file is used for Japanese language.
- *Resource.Ar.resx*- This resource file is used for Arabic language.



Step 8 - Go to Solution Explorer and right click on Models=>Add=>Class.



Step 9 - In the following figure, give a name to your model class. Here, our class name is “Registration”.

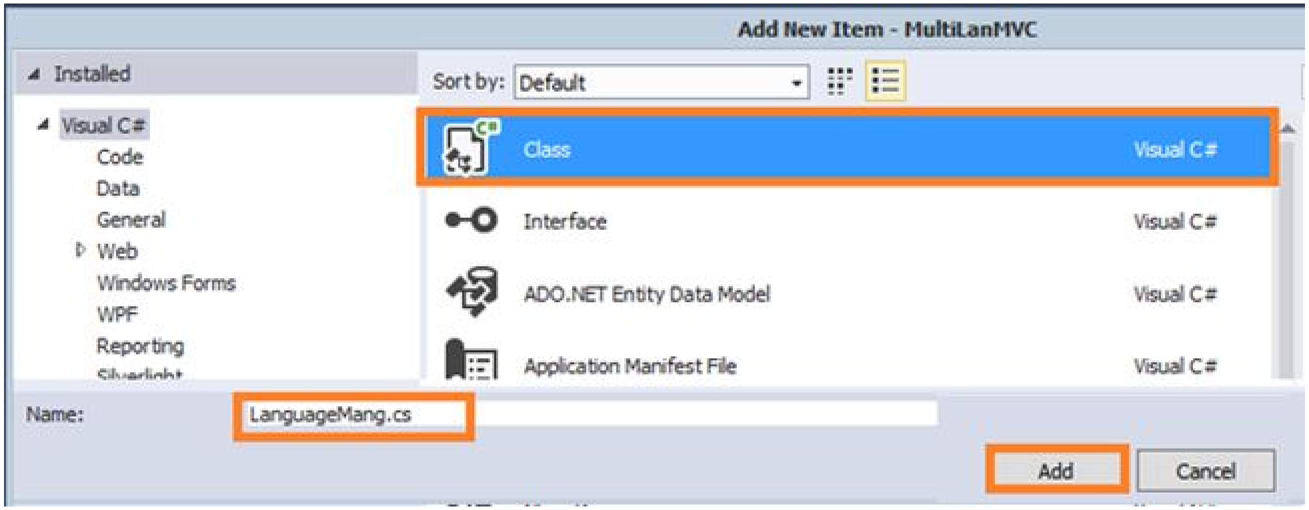


Name: Registration.cs

Our coding is given below-

```
System;
System.Collections.Generic;
System.ComponentModel.DataAnnotations;
System.Linq;
System.Web;
ace MultiLanMVC.Models

blic class Registration {
  [Display(Name = "FirstName", ResourceType = typeof(Resource))]
  [Required(ErrorMessageResourceType = typeof(Resource), ErrorMessageResourceName = "FirstNameRequired")]
  public string FirstName {
      get;
      set;
  }
  [Display(Name = "LastName", ResourceType = typeof(Resource))]
  [Required(ErrorMessageResourceType = typeof(Resource), ErrorMessageResourceName = "LastNameRequired")]
  public string LastName {
      get;
      set;
  }
  [Display(Name = "Email", ResourceType = typeof(Resource))]
  [Required(ErrorMessageResourceType = typeof(Resource), ErrorMessageResourceName = "EmailRequired")]
$", ErrorMessageResourceType = typeof(Resource), ErrorMessageResourceName = "EmailInvalid")]
  public string Email {
      get;
      set;
  }
  [Display(Name = "Country", ResourceType = typeof(Resource))]
  [Required(ErrorMessageResourceType = typeof(Resource), ErrorMessageResourceName = "CountryNameRequired")]
  public string Country {
      get;
      set;
  }
```

Step 10 - Go to Solution Explorer= > Right click on Project name= > Add= > Class => Enter Class name= > Add.

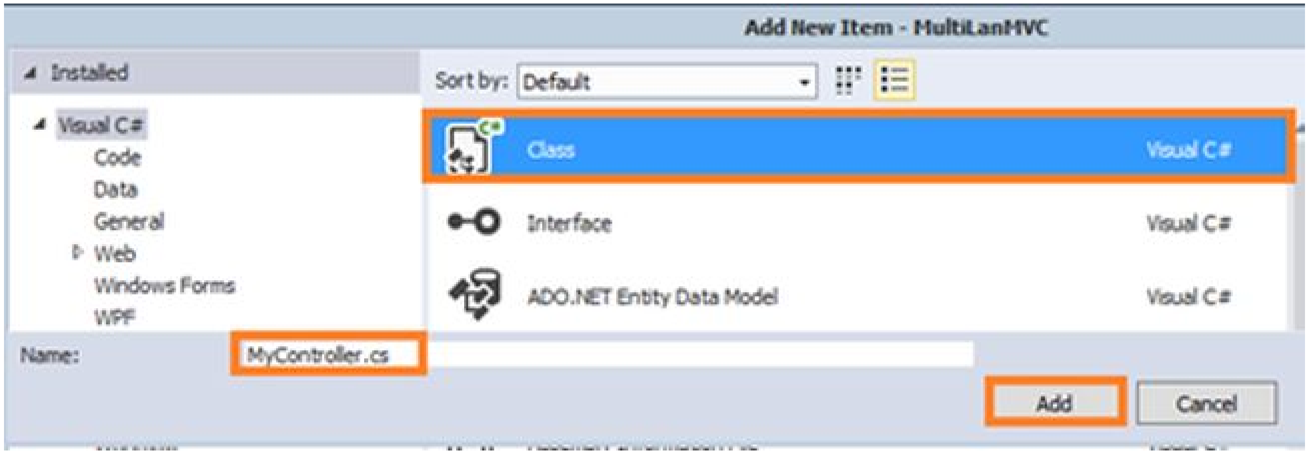


Name: LanguageMang.cs

Coding

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Web;
namespace MultiLanMVC
{
    public class LanguageMang {
        public static List < Languages > AvailableLanguages = new List < Languages > {
            new Languages {
                LanguageFullName = "English", LanguageCultureName = "en"
            },
            new Languages {
                LanguageFullName = "Hindi", LanguageCultureName = "Hi"
            },
            new Languages {
                LanguageFullName = "japanese", LanguageCultureName = "Ja"
            },
            new Languages {
                LanguageFullName = "arabic", LanguageCultureName = "Ar"
            },
        };
        public static bool IsLanguageAvailable(string lang) {
            return AvailableLanguages.Where(a => a.LanguageCultureName.Equals(lang)).FirstOrDefault()
        }
        public static string GetDefaultLanguage() {
            return AvailableLanguages[0].LanguageCultureName;
        }
        public void SetLanguage(string lang) {
            try {
                if (!IsLanguageAvailable(lang)) lang = GetDefaultLanguage();
                var cultureInfo = new CultureInfo(lang);
                Thread.CurrentThread.CurrentUICulture = cultureInfo;
                Thread.CurrentThread.CurrentCulture = CultureInfo.CreateSpecificCulture(cultureInfo.N
                HttpCookie langCookie = new HttpCookie("culture", lang);
                langCookie.Expires = DateTime.Now.AddYears(1);
                HttpContext.Current.Response.Cookies.Add(langCookie);
            } catch (Exception) {}
        }
    }
    public class Languages {
        public string LanguageFullName {
            get;
        public string LanguageCultureName {
            get;
            set;
        }
    }
}
```

Step 11 - Here, add another class. Go to Solution Explorer= > Right click on Project name= > Add= > Class => Enter class name= > Add. Here, our Class name is “MyController”.



Sort by: Default

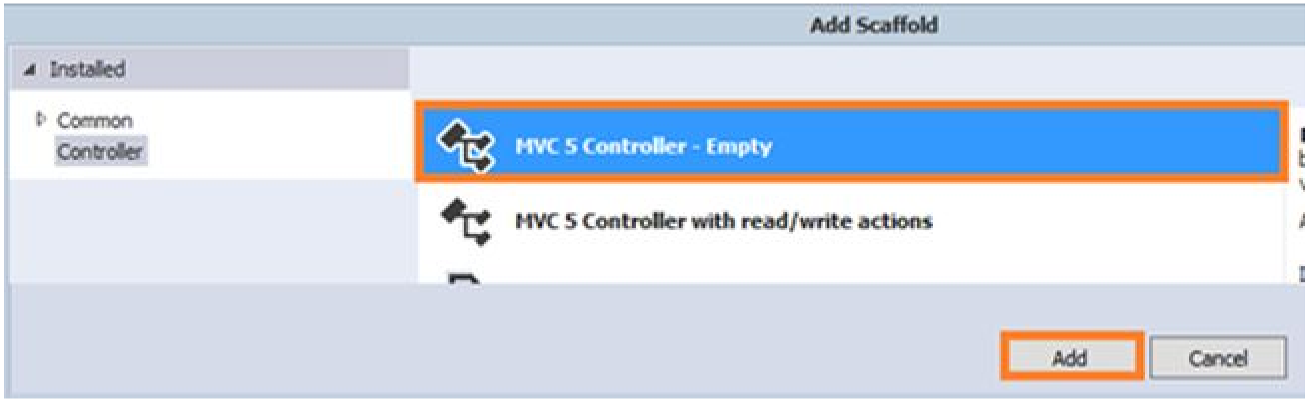
Name: MyController.cs

Our coding is given below-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
namespace MultiLanMVC
{
    public class MyController: Controller
    {
        protected override IAsyncResult BeginExecuteCore(AsyncCallback callback, object state) {
            string lang = null;
```

```
                lang = langCookie.Value;
            } else {
                var userLanguage = Request.UserLanguages;
                var userLang = userLanguage != null ? userLanguage[0] : "";
                if (userLang != "") {
                    lang = userLang;
                } else {
                    lang = LanguageMang.GetDefaultLanguage();
                }
            }
            new LanguageMang().SetLanguage(lang);
            return base.BeginExecuteCore(callback, state);
        }
    }
}
```

Step 12 - Now, right click on your Controller folder and add=>Controller=>Empty Controller and add Controller name. I have a controller name as “HomeController”.

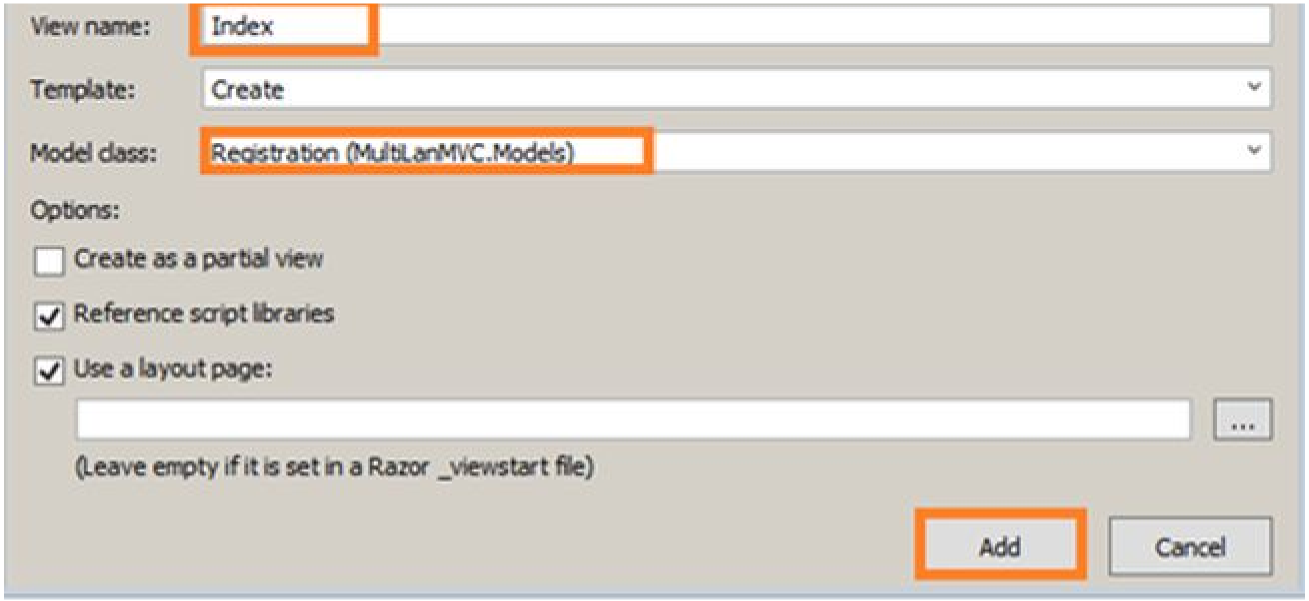


Code

```
using MultiLanMVC.Models;
using System;
using System.Collections.Generic;
using System.Linq;
```

```
namespace MultiLanMVC.Controllers
{
    public class HomeController: MyController {
        // GET: Home
        public ActionResult Index() {
                return View();
            }
            [HttpPost]
        public ActionResult Index(Registration r) {
            return View(r);
        }
        public ActionResult ChangeLanguage(string lang) {
            new LanguageMang().SetLanguage(lang);
            return RedirectToAction("Index", "Home");
        }
    }
}
```

Step 13 - Now, right click on ActionResult and add a new View. Click to add button.



View name: Index

Template: Create

Model class: Registration (MultiLanMVC.Models)

Options:

□ Create as a partial view

☑ Reference script libraries

☑ Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)

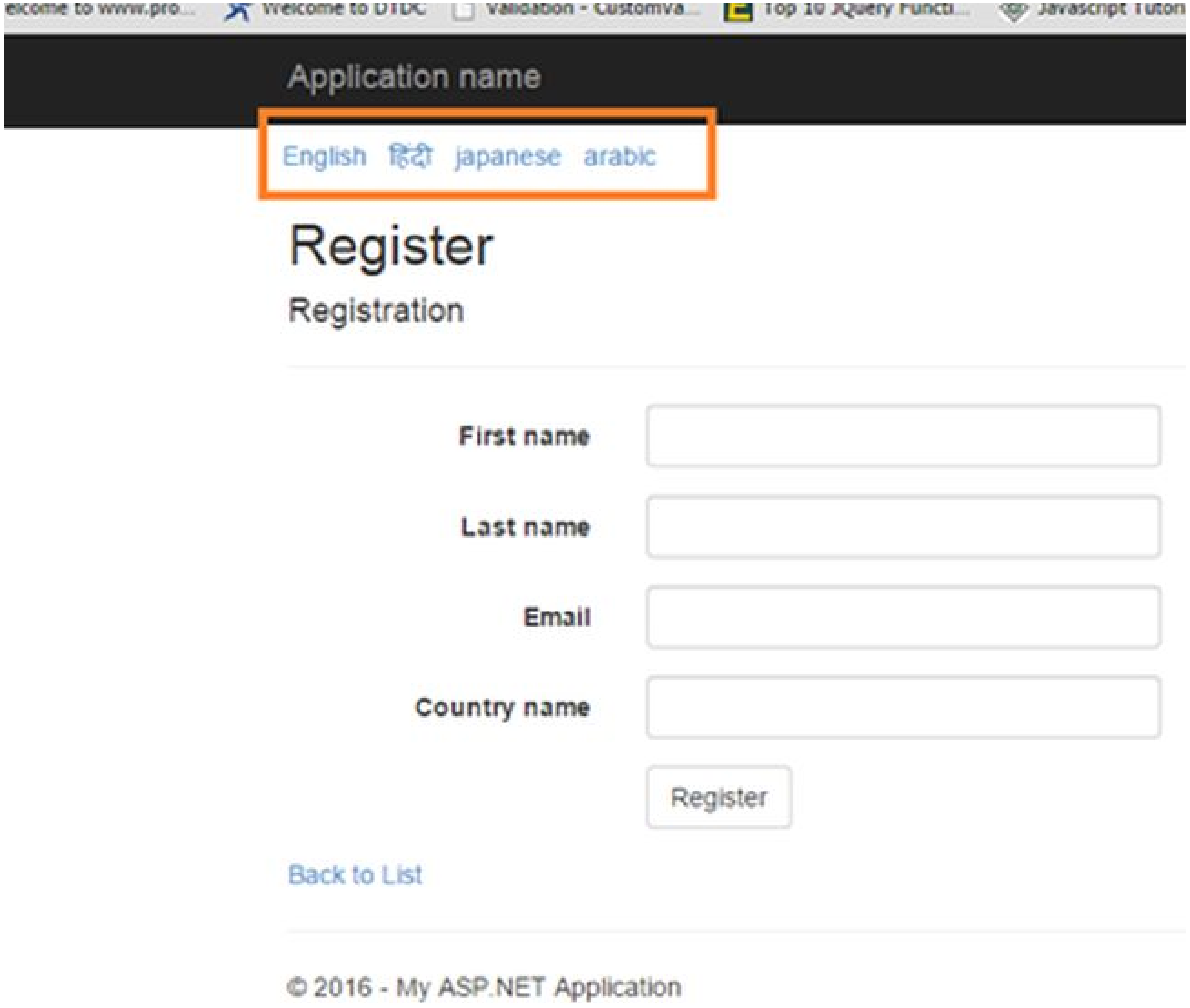
```
@model MultiLanMVC.Models.Registration
@ {
    ViewBag.Title = MultiLanMVC.Resource.Register;
} < h2 > @MultiLanMVC.Resource.Register < /h2>
@using(Html.BeginForm()) {
    @Html.AntiForgeryToken() < div class = "form-
horizontal" > < h4 > Registration < /h4>   < hr / > @Html.ValidationSummary(true, "", new {
        @class = "text-danger"
    }) < div class = "form-
group" > @Html.LabelFor(model => model.FirstName, htmlAttributes: new {
        @class = "control-label col-md-2"
    }) < div class = "col-md-10" > @Html.EditorFor(model => model.FirstName, new {
        htmlAttributes = new {
            @class = "form-control"
        }
    })
    @Html.ValidationMessageFor(model => model.FirstName, "", new {
        @class = "text-danger"
    }) < /div>   < /div>   < div class = "form-
group" > @Html.LabelFor(model => model.LastName, htmlAttributes: new {
        htmlAttributes = new {
            @class = "form-control"
        }
    })
    @Html.ValidationMessageFor(model => model.LastName, "", new {
        @class = "text-danger"
    }) < /div>   < /div>   < div class = "form-group" > @Html.LabelFor(model => model.Email, htmlAttributes: new {
        @class = "control-label col-md-2"
    }) < div class = "col-md-10" > @Html.EditorFor(model => model.Email, new {
        htmlAttributes = new {
            @class = "form-control"
        }
    })
    @Html.ValidationMessageFor(model => model.Email, "", new {
        @class = "text-danger"
    }) < /div>   < /div>   < div class = "form-group" > @Html.LabelFor(model => model.Country, htmlAttributes: new {
        @class = "control-label col-md-2"
    }) < div class = "col-md-10" > @Html.EditorFor(model => model.Country, new {
        htmlAttributes = new {
            @class = "form-control"
        }
    })
    @Html.ValidationMessageFor(model => model.Country, "", new {
        @class = "text-danger"
    }) < /div>   < /div>   < div class = "form-group" > < div class = "col-md-offset-2 col-md-10" > < input type = "submit"
    value = "@MultiLanMVC.Resource.Register"
    class = "btn btn-default" / > < /div>   < /div>   < /div>
} < div > @Html.ActionLink("Back to List", "Index") < /div>   < script src = "~/Scripts/jquery-1.10.2.min.js" > < /script>   < script src = "~/Scripts/jquery.validate.min.js" > < /script>   < scr
```

Step 14 - Add the given code in Complete Layout page. (_Layout .cshtml).

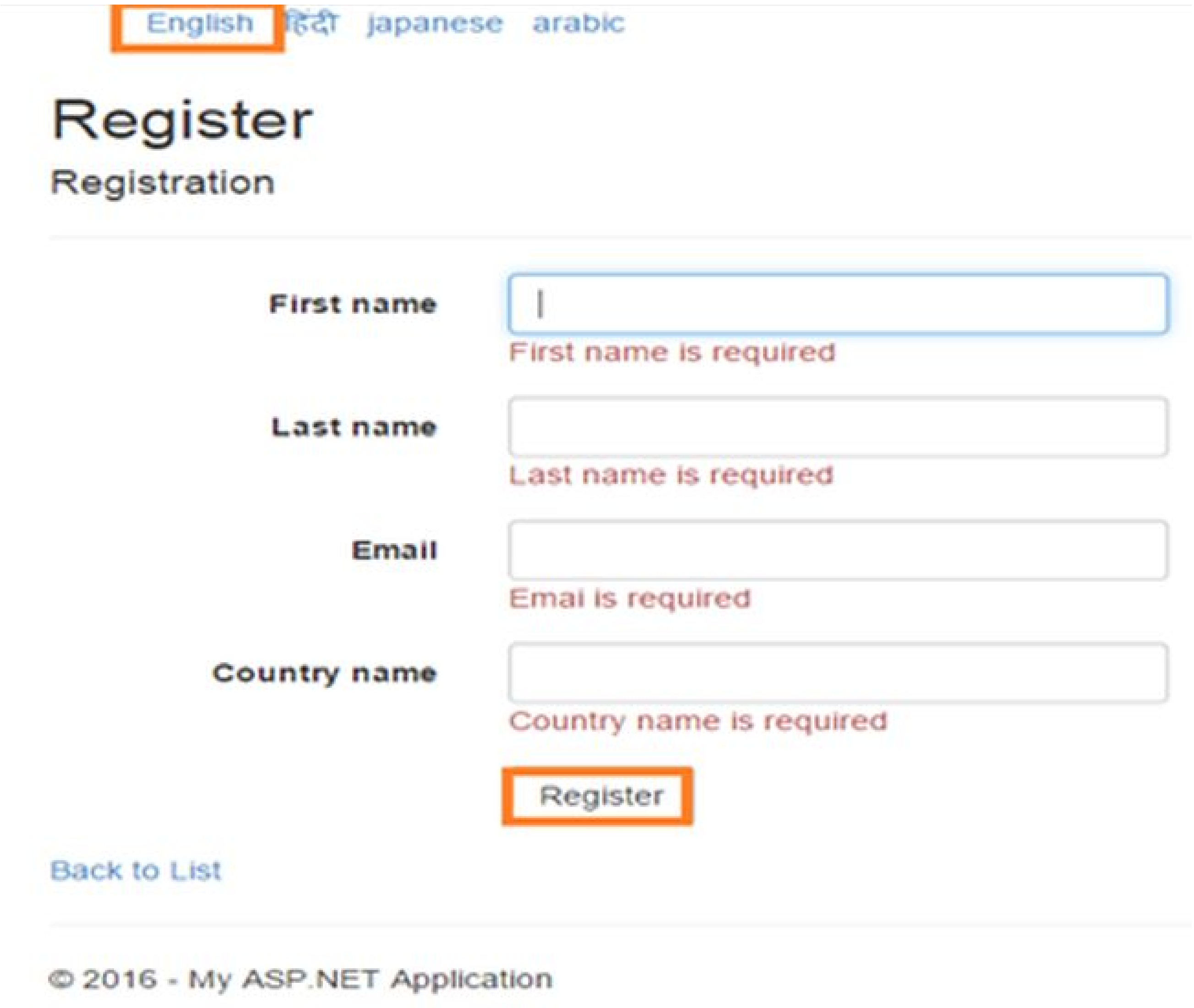
```
<div style="padding:5px">

@{
```

```
@Html.ActionLink(i.LanguageFullName, "ChangeLanguage", "Home", new { lang = i.LanguageCultureName },
}
}
</div>
```

Here, the default output is given blow-



Click English tab at the top of the page and you will see all the required validations are in Spanish language.



English हिंदी japanese arabic

Register

Registration

First name

First name is required

Last name

Last name is required

Email

Emai is required

Country name

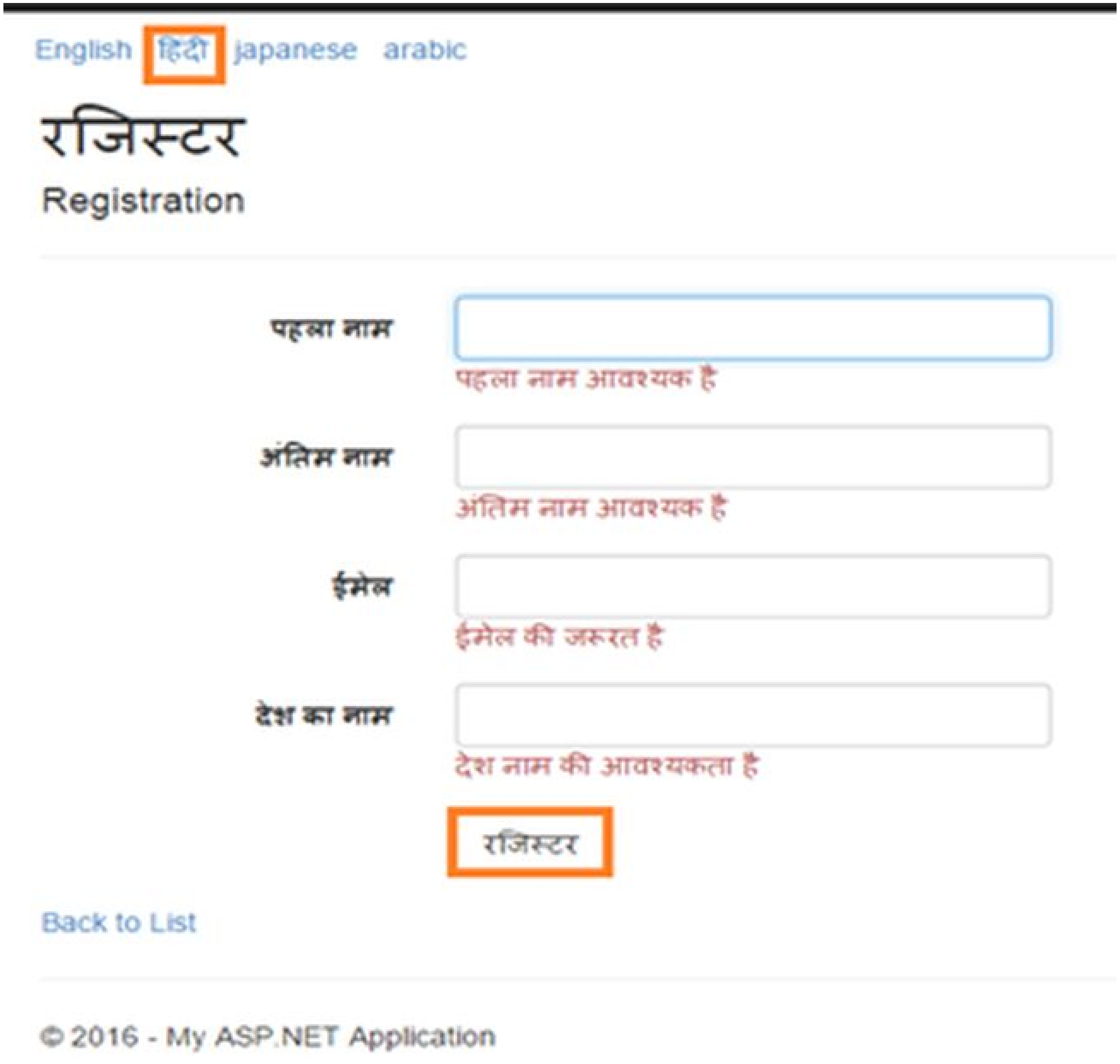
Country name is required

Register

Back to List

© 2016 - My ASP.NET Application

Click Hindi tab at the top of the page and then you will see all the required validations are in Hindi language.



रजिस्टर

Registration

पहला नाम

पहला नाम आवश्यक है

अंतिम नाम

अंतिम नाम आवश्यक है

ईमेल

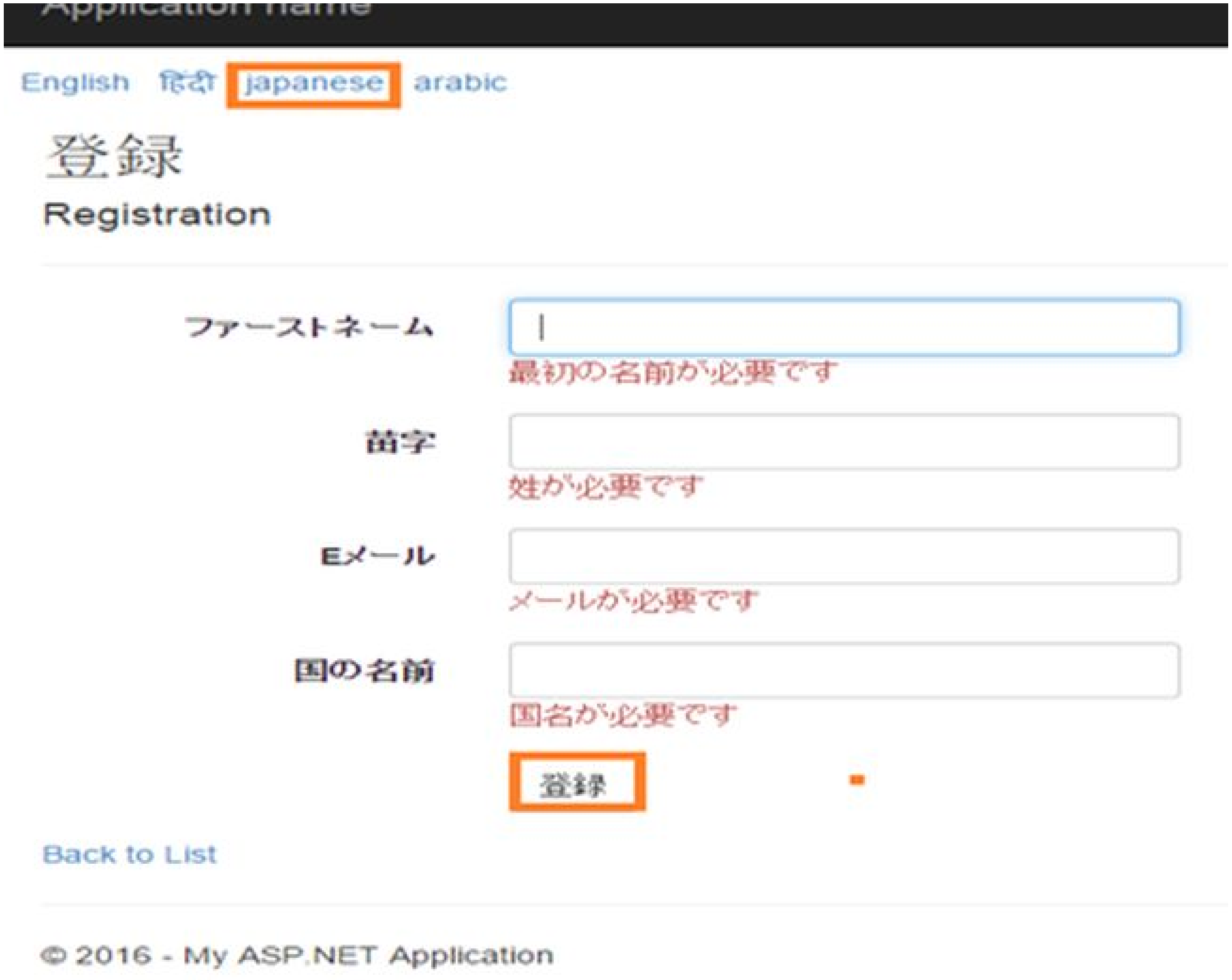
ईमेल की जरूरत है

देश का नाम

देश नाम की आवश्यकता है

© 2016 - My ASP.NET Application

Click Japanese tab at the top of the page and you will see all the required validations are in Japanese language.



登録

Registration

ファーストネーム

最初の名前が必要です

苗字

姓が必要です

Eメール

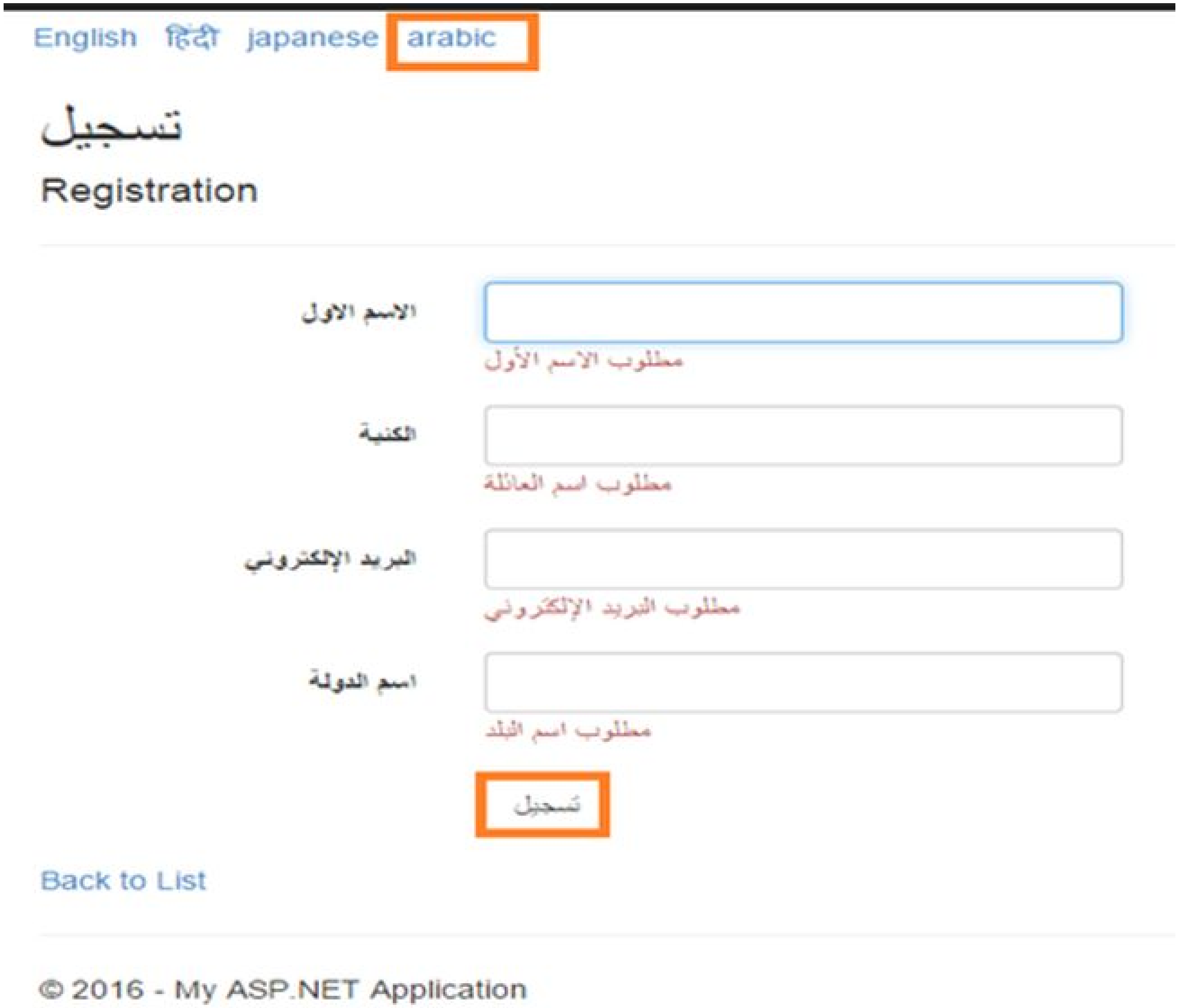
メールが必要です

国の名前

国名が必要です

© 2016 - My ASP.NET Application

Click Arabic tab at the top of the page and then you will see all the required validations are in Arabic language.



تسجيل

Registration

الاسم الاول

مطلوب الاسم الأول

الكنية

مطلوب اسم العائلة

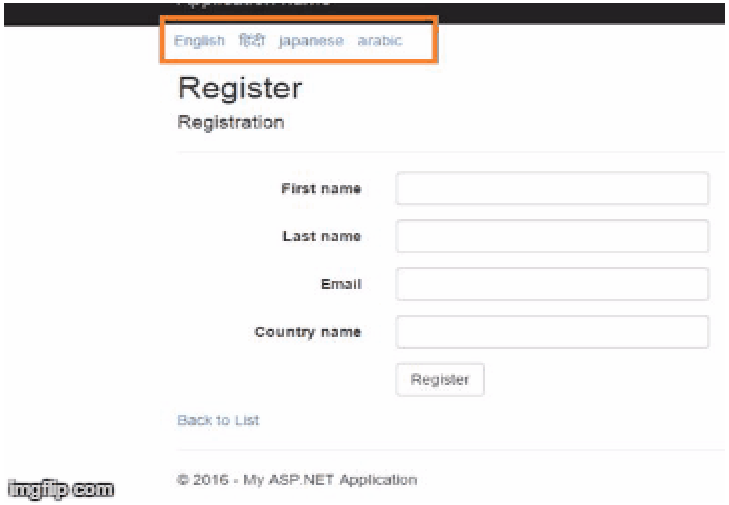
البريد الإلكتروني

مطلوب البريد الإلكتروني

اسم الدولة

مطلوب اسم البلد

© 2016 - My ASP.NET Application



English हिंदी japanese arabic

Register

Registration

First name

Last name

Email

Country name

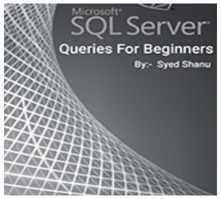
I hope you enjoyed this multi-language tutorial. Follow C# Corner to learn new things about ASP.NET MVC. Thanks for reading this article.

ASP.NET MVC Resources File Visual Studio

Next Recommended Reading

Upload Multiple Files Using FileUpload Control In ASP.NET 4.5

OUR BOOKS



Avinash Thakur

https://www.c-sharpcorner.com/members/avinash-thakur3

1209 **285.4k**

View Previous Comments

15 **39**

Type your comment here and press Enter Key (Minimum 10 characters)

Please share project's files.

ss duq

2028 **13** **0**

Jun 07, 2021

0 0 Reply

How to add error to resource file which we validate inside the controller. For eg: if drop down selected or not we add a validation then how to add that error message to resource file?

pushpa Acharya

1971 **70** **0**

Feb 02, 2021

0 0 Reply

In Registration class, typeof(Resource) shows error[Display(Name = "FirstName", ResourceType = typeof(Resource))]

Imtiaj Ahammad

2016 **26** **0**

Dec 29, 2020

0 1 Reply

In your solution default Resource.resx file should be present.

Only change my Datetime

Nk kanniyan

2030 11 167

Dec 08, 2020

0 0 Reply

Thanks for this tutorial, why i debug SetLanguage it goes to debug 2 times ?

Mahamod Goda

2001 40 0

Oct 29, 2020

0 0 Reply

Can you give me code for in .net core 2.1

suraj balwani

1893 148 0

Feb 11, 2020

0 0 Reply

It is not working for me when i am going to click on other lang it is displaying only english.

Aravind Kumar

1530 511 487

Jan 21, 2020

0 0 Reply

If the code didn't work for you, it's probably because you didn't enable the cookies in your Web.config.

Ibrahim Yassin

1949 92 0

Oct 22, 2019

0 1 Reply

Go To Web.config | inside the <system.web> add (<httpCookies httpOnlyCookies="true"/>).

Ibrahim Yassin

1949 92 0

Oct 22, 2019

0

Is there any need to add some code in global.cs

Shidhu (SP)

1967 74 0

Aug 16, 2019

0 0 Reply

It works for me, to make the tutorial work you need to make sure that for example in LanguageMang.cs : new Languages { LanguageFullName = "French", LanguageCultureName = "Fr"} your resource file is named : Resource.Fr.resx

abir attia

1987 54 0

Jul 30, 2019

0 1

1967 74 0

0

FEATURED ARTICLES

Understanding Synchronization Context Task.ConfigureAwait In Action

What is Non Fungible Tokens (NFT)? Why NFTs are So Popular Today?

Building Secure REST API

Unit Testing With xUnit And Moq In ASP.NET Core

Using Certificates For API Authentication In .NET 5

View All

TRENDING UP

01 Use Dynamic Data Masking To Protect Sensitive Data In Azure SQL Database

02 Dynamics 365 Solution Export & Import as Managed Using AzureDevOps Build & Release Pipeline

03 <⚡> Time Triggered Azure Functions - A Guide To Background Tasks Using C#

04 Agile Methodology In Nutshell

05 Some Cool Features In C# 10

06 Caching Mechanism In ASP.NET Core

07 Implementing Unit Of Work And Repository Pattern With Dependency Injection In .Net 5

08 Difference Between HAVING And WHERE Clause In SQL Server

09 Migrating From salesforce To Microsoft Dynamics 365

View All

About Us Contact Us Privacy Policy Terms Media Kit Sitemap Report a Bug FAQ Partners

C# Tutorials Common Interview Questions Stories Consultants Ideas Certifications