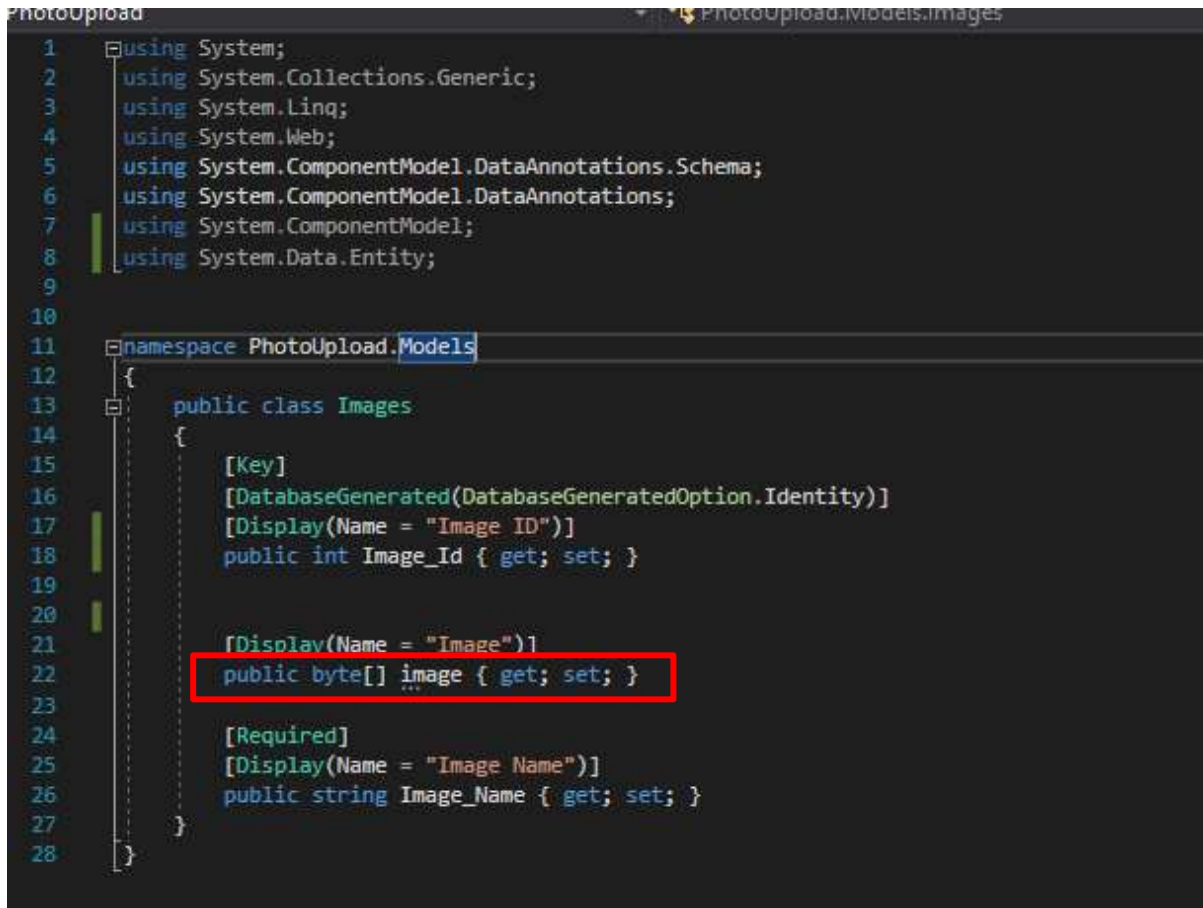


STEP1

- Create a class (Images)
- Don't forget to include this.



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.ComponentModel.DataAnnotations.Schema;
6 using System.ComponentModel.DataAnnotations;
7 using System.ComponentModel;
8 using System.Data.Entity;
9
10
11 namespace PhotoUpload.Models
12 {
13     public class Images
14     {
15         [Key]
16         [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
17         [Display(Name = "Image ID")]
18         public int Image_Id { get; set; }
19
20
21         [Display(Name = "Image")]
22         public byte[] image { get; set; }
23
24         [Required]
25         [Display(Name = "Image Name")]
26         public string Image_Name { get; set; }
27     }
28 }
```

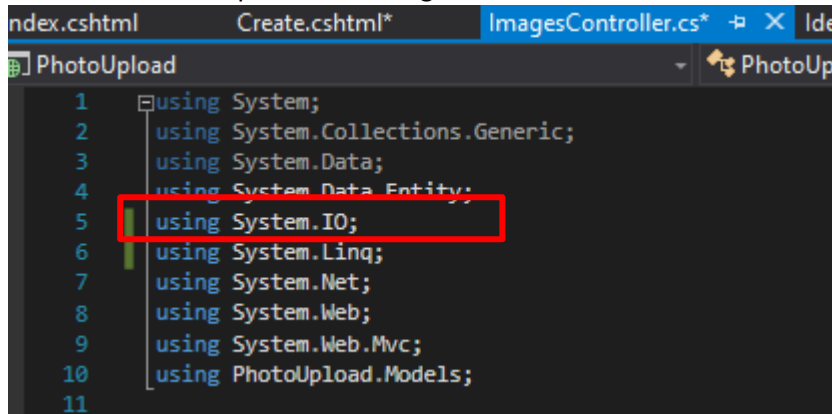
```
namespace PhotoUpload.Models
{
    public class Images
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        [Display(Name = "Image ID")]
        public int Image_Id { get; set; }

        [Display(Name = "Image")]
        public byte[] image { get; set; }

        [Required]
        [Display(Name = "Image Name")]
        public string Image_Name { get; set; }
    }
}
```

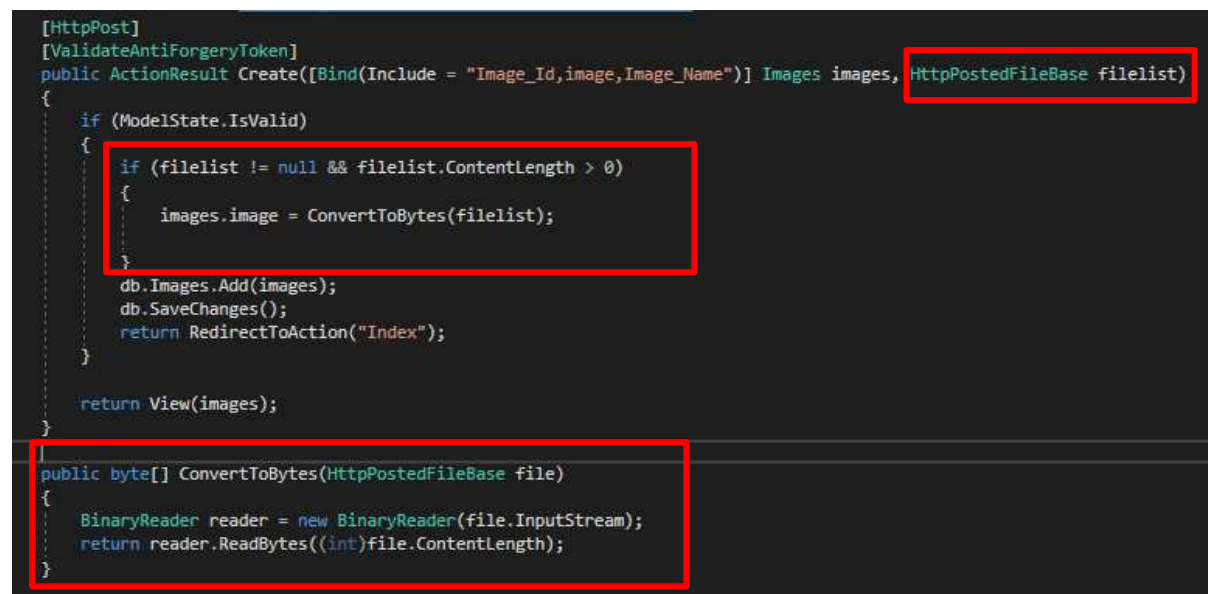
STEP2

- Scaffold the controller
- include this namespace in the imagesController.



```
1 using System;
2 using System.Collections.Generic;
3 using System.Data;
4 using System.Data.Entity;
5 using System.IO;
6 using System.Linq;
7 using System.Net;
8 using System.Web;
9 using System.Web.Mvc;
10 using PhotoUpload.Models;
11
```

Include the following code on Create() method.



```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "Image_Id,image,Image_Name")] Images images, HttpPostedFileBase filelist)
{
    if (ModelState.IsValid)
    {
        if (filelist != null && filelist.ContentLength > 0)
        {
            images.image = ConvertToBytes(filelist);
        }
        db.Images.Add(images);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    return View(images);
}

public byte[] ConvertToBytes(HttpPostedFileBase file)
{
    BinaryReader reader = new BinaryReader(file.InputStream);
    return reader.ReadBytes((int)file.ContentLength);
}
```

```

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "Image_Id,image,Image_Name")]
Images images, HttpPostedFileBase filelist)
{
    if (ModelState.IsValid)
    {
        if (filelist != null && filelist.ContentLength > 0)
        {
            images.image = ConvertToBytes(filelist);
        }
        db.Images.Add(images);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    return View(images);
}

public byte[] ConvertToBytes(HttpPostedFileBase file)
{
    BinaryReader reader = new BinaryReader(file.InputStream);
    return reader.ReadBytes((int)file.ContentLength);
}

```

STEP3

- Go to views > images > Create.cshtml
- Add and edit with the following code.

```

@using (Html.BeginForm("Create", "Images", FormMethod.Post, new { enctype = "multipart/form-data" }))
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Images</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })

        <div class="form-group">
            @Html.LabelFor(model => model.image, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-3">
                <div>
                    <input type="file" id="filelist" name="filelist" />
                </div>
                @Html.ValidationMessageFor(model => model.image, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Image_Name, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Image_Name, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Image_Name, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">

```

```
@model PhotoUpload.Models.Images
```

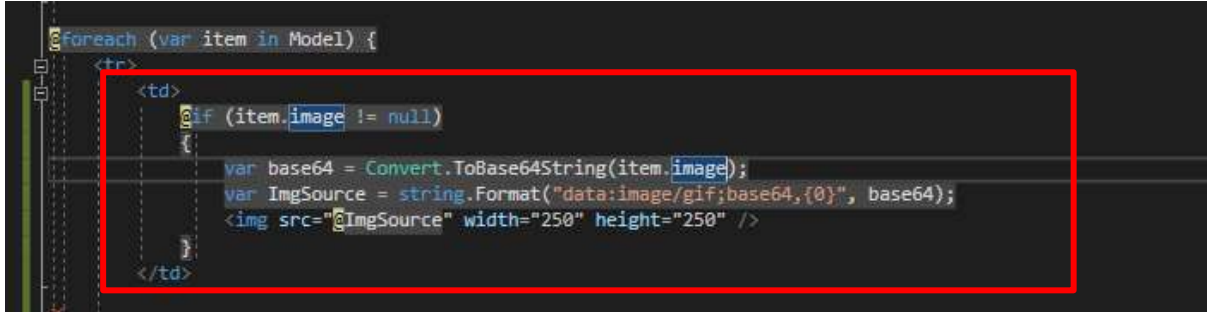
```
@{  
    ViewBag.Title = "Create";  
}
```

```
<h2>Create</h2>
```

```
@using (Html.BeginForm("Create", "Images", FormMethod.Post, new { enctype =  
    "multipart/form-data" }))  
{  
    @Html.AntiForgeryToken()  
  
    <div class="form-horizontal">  
        <h4>Images</h4>  
        <hr />  
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })  
  
        <div class="form-group">  
            @Html.LabelFor(model => model.image, htmlAttributes: new { @class =  
                "control-label col-md-2" })  
            <div class="col-md-3">  
                <div>  
                    <input type="file" id="filelist" name="filelist" />  
                </div>  
                @Html.ValidationMessageFor(model => model.image, "", new { @class  
                    = "text-danger" })  
            </div>  
        </div>  
  
        <div class="form-group">  
            @Html.LabelFor(model => model.Image_Name, htmlAttributes: new { @class  
                = "control-label col-md-2" })  
            <div class="col-md-10">  
                @Html.EditorFor(model => model.Image_Name, new { htmlAttributes =  
                    new { @class = "form-control" } })  
                @Html.ValidationMessageFor(model => model.Image_Name, "", new {  
                    @class = "text-danger" })  
            </div>  
        </div>  
  
        <div class="form-group">  
            <div class="col-md-offset-2 col-md-10">  
                <input type="submit" value="Create" class="btn btn-default" />  
            </div>  
        </div>  
    </div>  
}  
  
<div>  
    @Html.ActionLink("Back to List", "Index")  
</div>  
  
@section Scripts {  
    @Scripts.Render("~/bundles/jqueryval")  
}
```

STEP 4

- Go to views > images > Index.cshtml
- Edit with the following code.



```
@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.image)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Image_Name)
        </th>
        <th></th>
    </tr>

    @foreach (var item in Model) {
        <tr>
            <td>
                @if (item.image != null)
                {
                    var base64 = Convert.ToBase64String(item.image);
                    var ImgSource = string.Format("data:image/gif;base64,{0}", base64);
                    
                }
            </td>

            <td>
                @Html.DisplayFor(modelItem => item.Image_Name)
            </td>
            <td>
                @Html.ActionLink("Edit", "Edit", new { id = item.Image_Id }) |
                @Html.ActionLink("Details", "Details", new { id = item.Image_Id }) |
                @Html.ActionLink("Delete", "Delete", new { id = item.Image_Id })
            </td>
        </tr>
    }
</table>
```

RESULTS

Application name

Home

About

Contact

Create

Images

Image

Choose File

No file chosen

Image Name

Create

[Back to List](#)

© 2020 - My ASP.NET Application

Index

[Create New](#)

Image	Image Name	
	TestImage	Edit Details Delete

© 2020 - My ASP.NET Application