

Проект Phones. Авторизация, аутентификация

Идея проекта:

Имеются данные о фирмах и телефонах. О фирмах известно: **название, страна**. О телефонах известно **название, цена, id фирмы**.

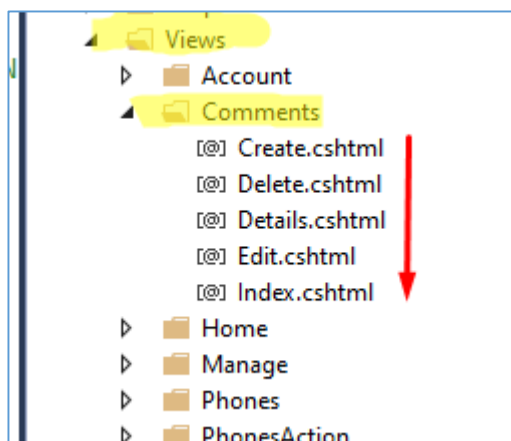
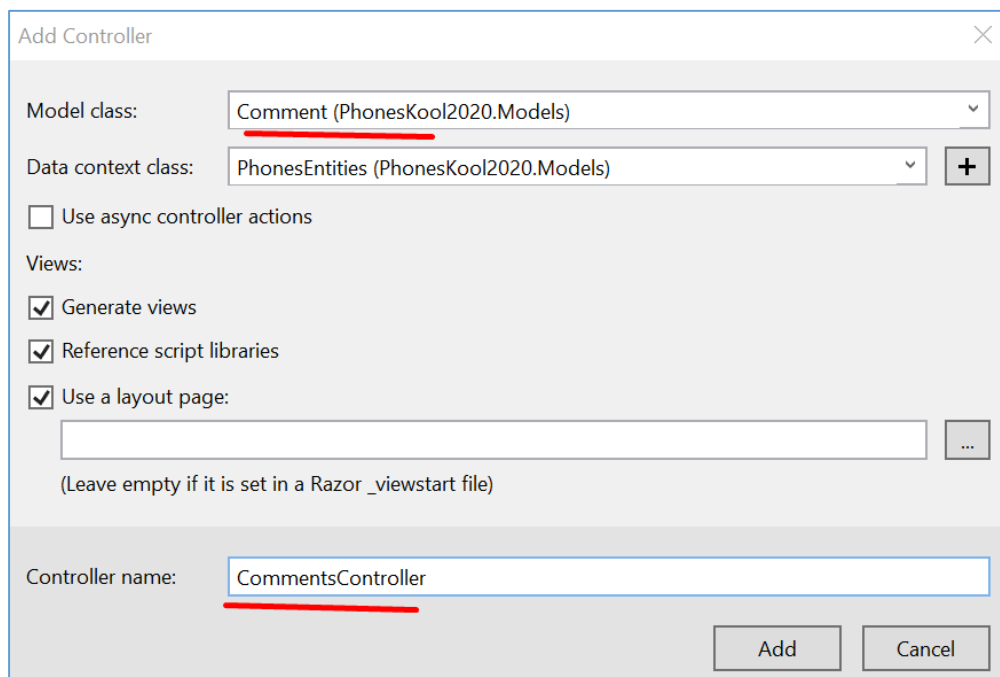
Данные хранятся в базе данных - информация о фирмах, телефонах.

Часть 5. Блок Комментарии

Зарегистрированные пользователи (client) могут добавить комментарии о продукте. **Админ** добавляет и удаляет комментарии – свои и других пользователей. Данные записываются в БД. Гостям добавим функционал - *гости читают комментарии*

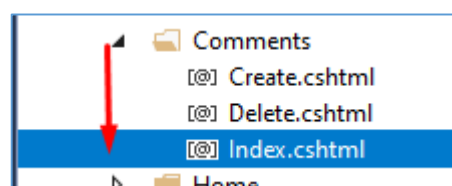
Шаг 1.

Создаем controller – **CommentsController** -> **Entity Framework (CRUD)** на основе модели **Comment**



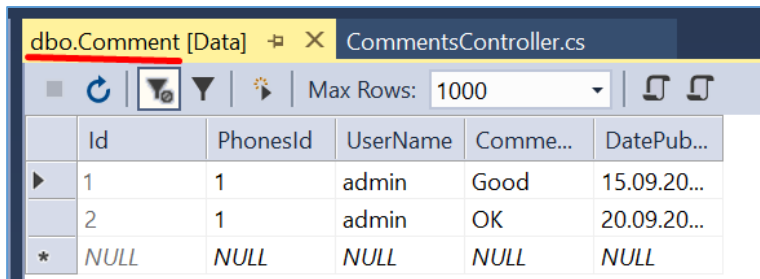
У нас создается полная структура для управления комментариями. Из всего списка нам понадобятся файлы **Index, Create, Delete**

Detail, Edit можно удалить.



Шаг 2.

Выведем список комментариев к продуктам (если они введены). У нас добавлены комментарии к продукту с **Id=1** и добавлены админом



	Id	PhonesId	UserName	Comme...	DatePub...
▶	1	1	admin	Good	15.09.20...
	2	1	admin	OK	20.09.20...
*	NULL	NULL	NULL	NULL	NULL

Поскольку комментарии будут

- **добавляться** (авторизованными пользователями) - необходимо создать асинхронное обновление. Необходима форма добавления
- могут **удаляться** админом - необходимо асинхронное обновление

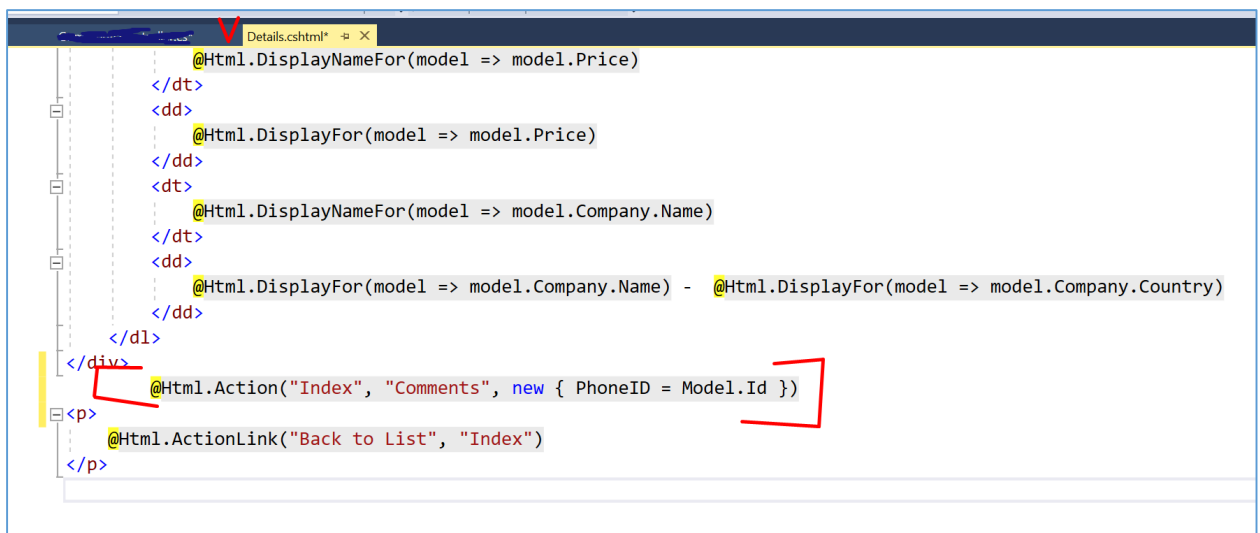
Список комментарии нужно **встроить** в лист **Details** нашего продукта – как частичное представление **PartialView**.

Метод **Index** объявим дочерним [**ChildActionOnly**]. Дочерние методы не могут быть вызваны в адресной строке.

Все методы для работы с комментариями будут **PartialViewResult**

Откройте файл **Views/Phones/Details.cshtml** – создадим ссылку для вывода комментариев

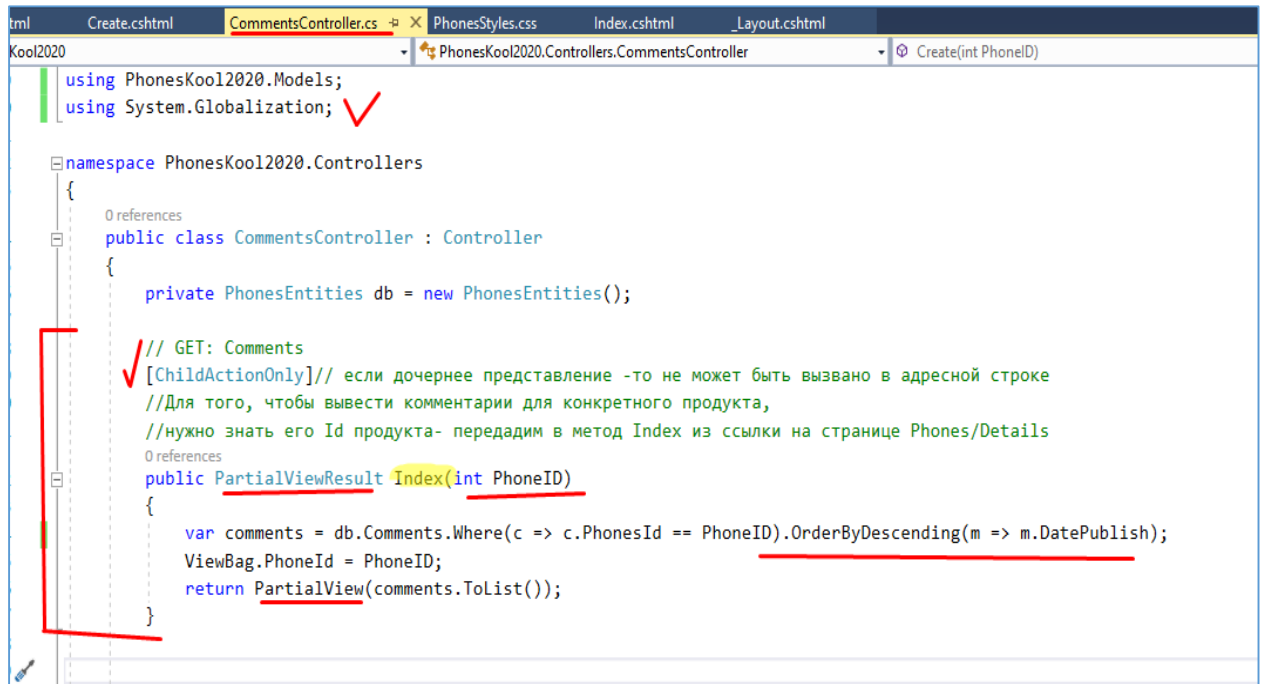
Используя хелпер **.Action** – встраиваем одно представление в другое. (как include в php)



```
@Html.DisplayNameFor(model => model.Price)
</dt>
<dd>
    @Html.DisplayFor(model => model.Price)
</dd>
<dt>
    @Html.DisplayNameFor(model => model.Company.Name)
</dt>
<dd>
    @Html.DisplayFor(model => model.Company.Name) - @Html.DisplayFor(model => model.Company.Country)
</dd>
</dl>
</div>
@Html.Action("Index", "Comments", new { PhoneID = Model.Id })
<p>
    @Html.ActionLink("Back to List", "Index")
</p>
```

Контроллер **CommentsController** метод **Index**

Заменяем код:



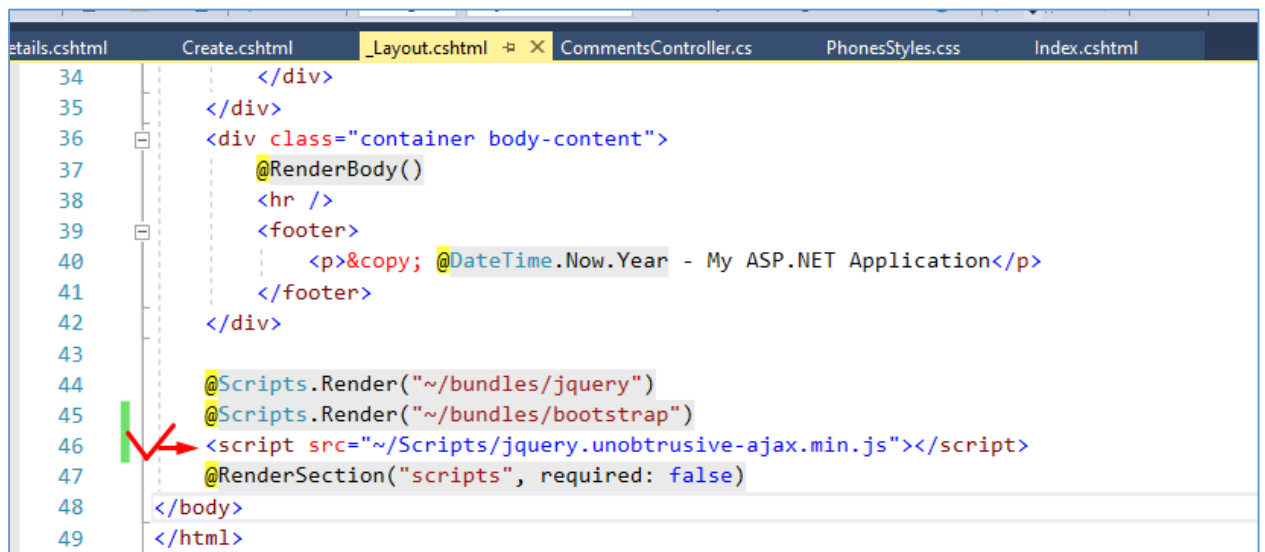
```
using PhonesKool2020.Models;
using System.Globalization; ✓

namespace PhonesKool2020.Controllers
{
    0 references
    public class CommentsController : Controller
    {
        private PhonesEntities db = new PhonesEntities();

        // GET: Comments
        ✓ [ChildActionOnly] // если дочернее представление -то не может быть вызвано в адресной строке
        //Для того, чтобы вывести комментарии для конкретного продукта,
        //нужно знать его Id продукта- передадим в метод Index из ссылки на странице Phones/Details
        0 references
        public PartialViewResult Index(int PhoneID)
        {
            var comments = db.Comments.Where(c => c.PhonesId == PhoneID).OrderByDescending(m => m.DatePublish);
            ViewBag.PhoneId = PhoneID;
            return PartialView(comments.ToList());
        }
    }
}
```

Файл **Views/Shared/_Layout.cshtml**

Добавим ссылку на скрипт **ajax** в конец кода



```
34         </div>
35     </div>
36     <div class="container body-content">
37         @RenderBody()
38         <hr />
39         <footer>
40             <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
41         </footer>
42     </div>
43
44     @Scripts.Render("~/bundles/jquery")
45     @Scripts.Render("~/bundles/bootstrap")
46     ✓ <script src="~/Scripts/jquery.unobtrusive-ajax.min.js"></script>
47     @RenderSection("scripts", required: false)
48 </body>
49 </html>
```

`<script src="~/Scripts/jquery.unobtrusive-ajax.min.js"></script>`

Изменим представление **Views/Comments/Index.cshtml**

```

1  @model IEnumerable<PhonesKool2020.Models.Comment>
2
3  @{
4      //асинхронное добавление-удаление комментариев
5      AjaxOptions ajaxOptions = new AjaxOptions { UpdateTargetId = "comments-tool" };
6  }
7  @using (Ajax.BeginForm("Index", new { PhoneId = ViewBag.PhoneId }, ajaxOptions))
8  {
9      <div id="comments-tool">
10         <div id="all-comments">
11             <hr />
12             <h3>Comments List</h3>
13             @foreach (var item in Model)//список комментариев по id товара
14             {
15                 <div class="phones-comment">
16                     <div class="phones-comment-title">
17                         From: @Html.DisplayFor(modelItem => item.UserName)
18                     </div>
19                     <div class="phones-comment-date">
20                         ✓ Date Publish: @item.DatePublish.Value.ToString("dd/MM/yyyy")
21                         <br />
22                         ✓ Date Publish: @item.DatePublish ← выбрать один из вариантов
23                     </div>
24                     <div class="phones-comment-body">
25                         @Html.DisplayFor(modelItem => item.Comments)
26                     </div>
27                     @if (Request.IsAuthenticated && User.IsInRole("admin"))
28                     {
29                         //если авторизовался admin ← удаляет только админ
30                         @Html.ActionLink("Delete This Comment", "Delete", new { id = item.Id })
31                     }
32                 </div>
33             }
34         </div>
35     }

```

```

7
8     <div id="addcomment" class="add-comment-box">
9         @if (Request.IsAuthenticated)
10         {
11             //если авторизовался пользователь - доступно добавить comment
12             @Html.Action("Create", "Comments", new { PhoneId = ViewBag.PhoneId })
13         }
14         else
15         {
16             @Html.ActionLink("Авторизуйтесь для добавления комментариев", "Login", "Account")
17         }
18     </div>
19
20 </div>
21 }

```

ПРОДОЛЖЕНИЕ КОДА

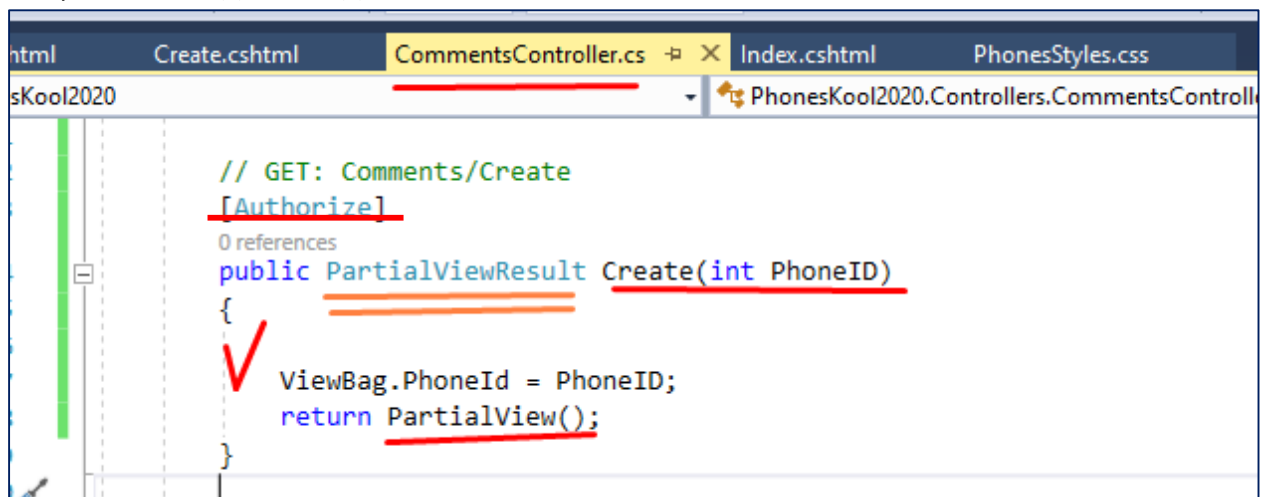
Code **Views/Comments/Index.cshtml**

```
@model IEnumerable<PhonesKool2020.Models.Comment>

@{
    //асинхронное добавление-удаление комментариев
    AjaxOptions ajaxOptions = new AjaxOptions { UpdateTargetId = "comments-tool" };
}
@using (Ajax.BeginForm("Index", new { PhoneId = ViewBag.PhoneId }, ajaxOptions))
{
    <div id="comments-tool">
        <div id="all-comments">
            <hr />
            <h3>Comments List</h3>
            @foreach (var item in Model)//список комментариев по id товара
            {
                <div class="phones-comment">
                    <div class="phones-comment-title">
                        From: @Html.DisplayFor(modelItem => item.UserName)
                    </div>
                    <div class="phones-comment-date">
                        Date Publish: @item.DatePublish.Value.ToString("dd/MM/yyyy")
                        <br />
                        Date Publish: @item.DatePublish
                    </div>
                    <div class="phones-comment-body">
                        @Html.DisplayFor(modelItem => item.Comments)
                    </div>
                    @if (Request.IsAuthenticated && User.IsInRole("admin"))
                    {
                        //если авторизовался admin
                        @Html.ActionLink("Delete This Comment", "Delete", new { id = item.Id
                    })
                }
            </div>
        }
    </div>
    <div id="addcomment" class="add-comment-box">
        @if (Request.IsAuthenticated)
        {
            //если авторизовался пользователь - доступно добавить comment
            @Html.Action("Create", "Comments", new { PhoneId = ViewBag.PhoneId })
        }
        else
        {
            @Html.ActionLink("Авторизуйтесь для добавления комментариев", "Login",
"Account")
        }
    </div>
</div>
}
```

Файл **Controller/CommentsController**

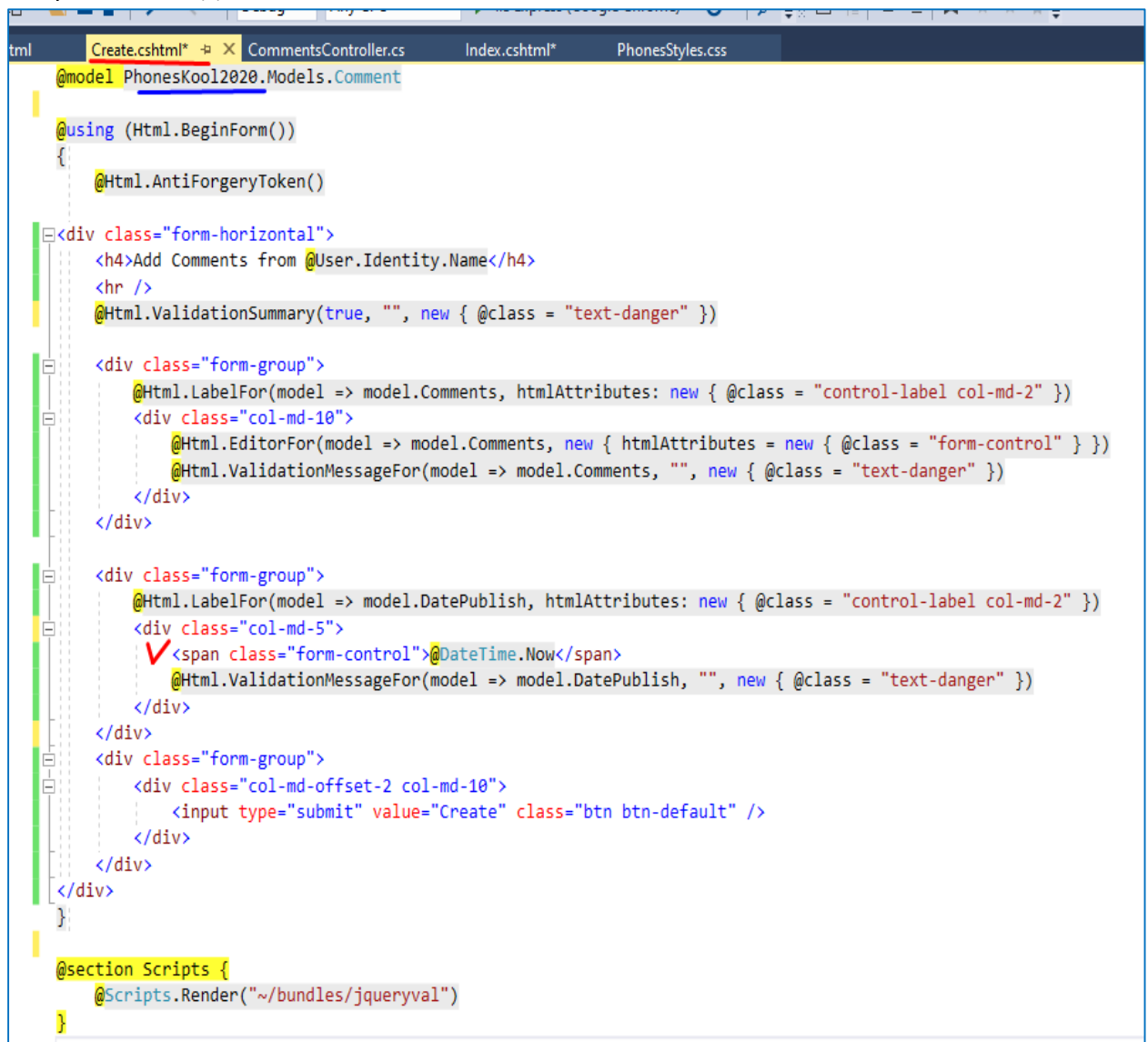
Исправляем код – метод **Create**



```
// GET: Comments/Create
[Authorize]
0 references
public PartialViewResult Create(int PhoneID)
{
    ViewBag.PhoneId = PhoneID;
    return PartialView();
}
```

Файл **Views/Comments/Create.cshtml**

Исправляем код



```
@model PhonesKool2020.Models.Comment
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Add Comments from @User.Identity.Name</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })

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

        <div class="form-group">
            @Html.LabelFor(model => model.DatePublish, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-5">
                <span class="form-control">@DateTime.Now</span>
                @Html.ValidationMessageFor(model => model.DatePublish, "", 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>

    @section Scripts {
        @Scripts.Render("~/bundles/jqueryval")
    }
}
```

Code Views/Comments/Create.cshtml

```
@model PhonesKool2020.Models.Comment

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Add Comments from @User.Identity.Name</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })

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

        <div class="form-group">
            @Html.LabelFor(model => model.DatePublish, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-5">
                <span class="form-control">@DateTime.Now</span>
                @Html.ValidationMessageFor(model => model.DatePublish, "", 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>
}

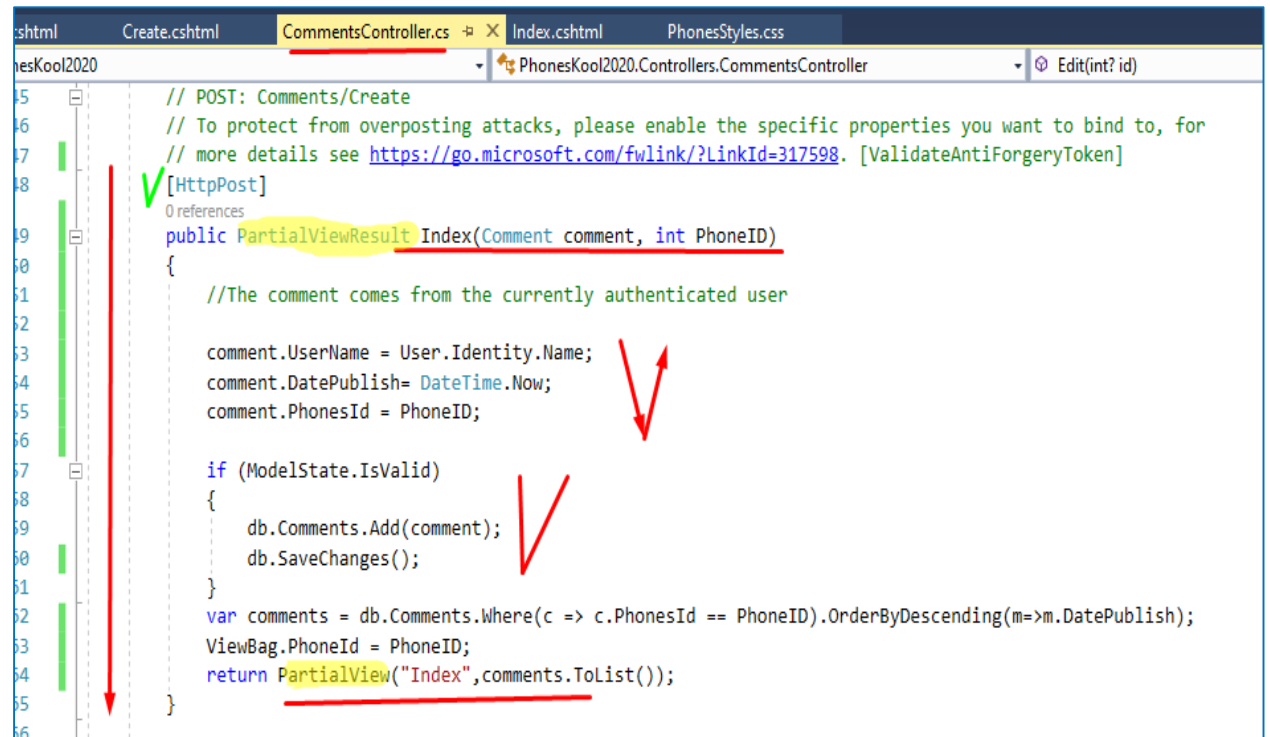
@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

После заполнения формы **Add Comment** возврат в **Controller**. И поскольку это частичное представление, встроенное в **Details** от метода **Index**, то метод возврата пишется по –другому

Название метод получает **Index** – с пометкой **[HttpPost]** и с типом **PartialViewResult**

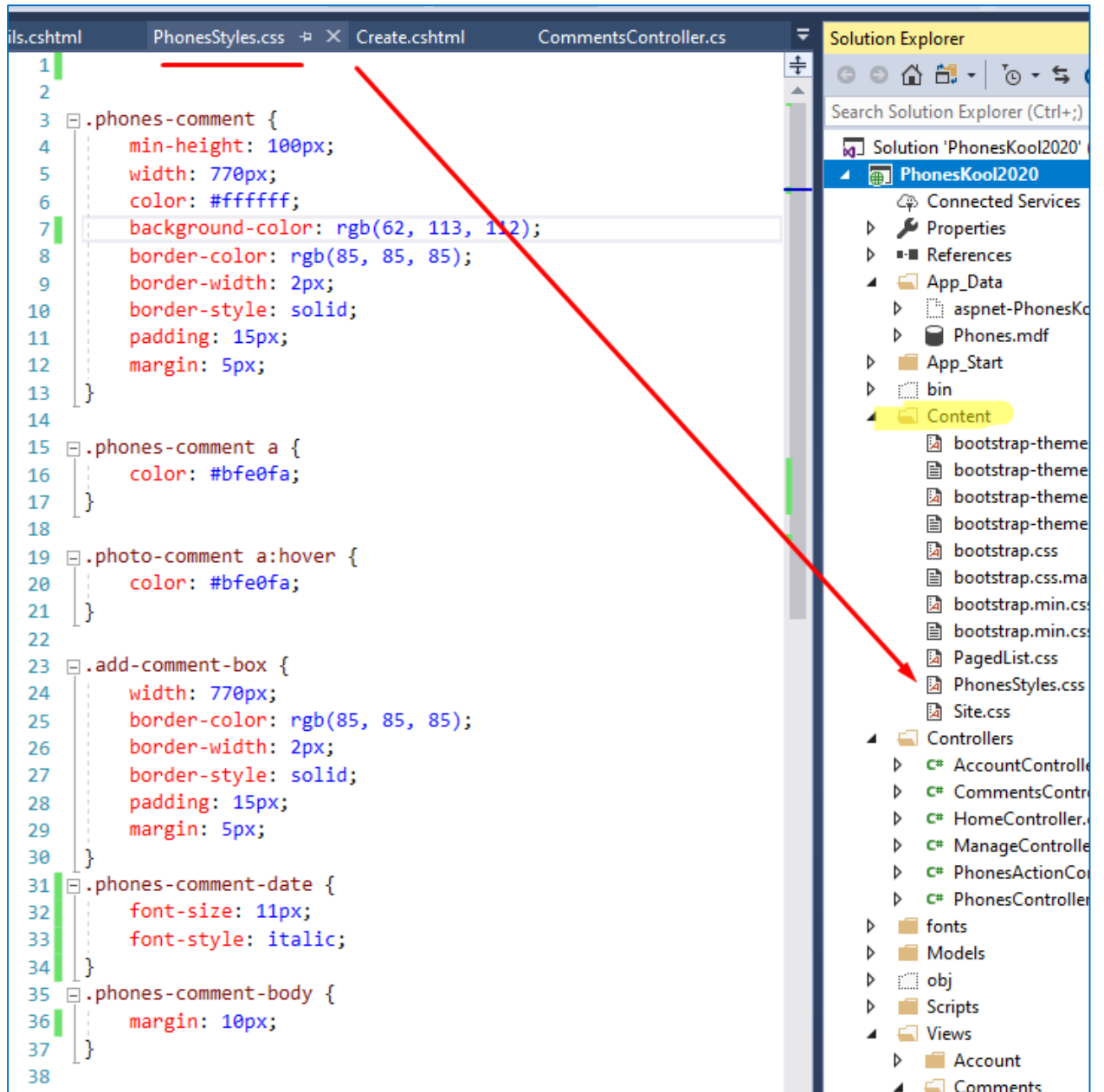
Файл Controller/CommentsController

Исправляем код – метод [HttpPost] **Create** меняем на [HttpPost] **Index**



```
15 // POST: Comments/Create
16 // To protect from overposting attacks, please enable the specific properties you want to bind to, for
17 // more details see https://go.microsoft.com/fwlink/?LinkId=317598. [ValidateAntiForgeryToken]
18 [HttpPost]
19 public PartialViewResult Index(Comment comment, int PhoneID)
20 {
21     //The comment comes from the currently authenticated user
22
23     comment.UserName = User.Identity.Name;
24     comment.DatePublish= DateTime.Now;
25     comment.PhonesId = PhoneID;
26
27     if (ModelState.IsValid)
28     {
29         db.Comments.Add(comment);
30         db.SaveChanges();
31     }
32     var comments = db.Comments.Where(c => c.PhonesId == PhoneID).OrderByDescending(m=>m.DatePublish);
33     ViewBag.PhoneId = PhoneID;
34     return PartialView("Index",comments.ToList());
35 }
```


Добавим стили для оформления вывода комментариев и формы **Add Comment**



Content/PhonesStyle.css

```
.phones-comment {
    min-height: 100px;
    width: 770px;
    color: #ffffff;
    background-color: rgb(62, 113, 112);
    border-color: rgb(85, 85, 85);
    border-width: 2px;
    border-style: solid;
    padding: 15px;
    margin: 5px;
}

.phones-comment a {
    color: #bfe0fa;
}

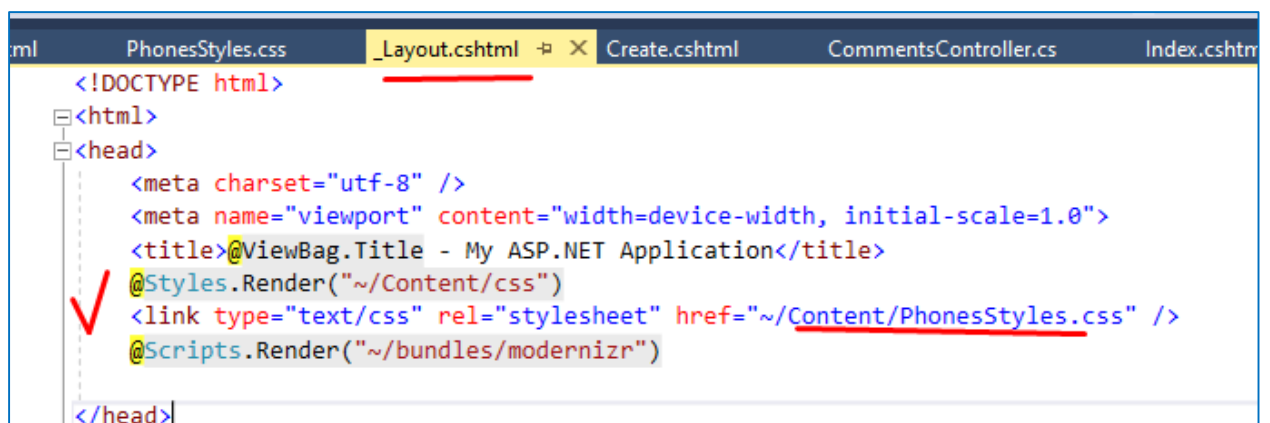
.photo-comment a:hover {
    color: #bfe0fa;
}

.add-comment-box {
    width: 770px;
    border-color: rgb(85, 85, 85);
    border-width: 2px;
    border-style: solid;
    padding: 15px;
    margin: 5px;
}

.phones-comment-date {
    font-size: 11px;
    font-style: italic;
}

.phones-comment-body {
    margin: 10px;
}
```

Подключим стили в файле **Views/Shared/_Layout.cshtml**



The screenshot shows the Visual Studio editor with the `_Layout.cshtml` file open. The code in the `<head>` section includes the following lines:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - My ASP.NET Application</title>
    @Styles.Render("~/Content/css")
    <link type="text/css" rel="stylesheet" href="~/Content/PhonesStyles.css" />
    @Scripts.Render("~/bundles/modernizr")
</head>
```

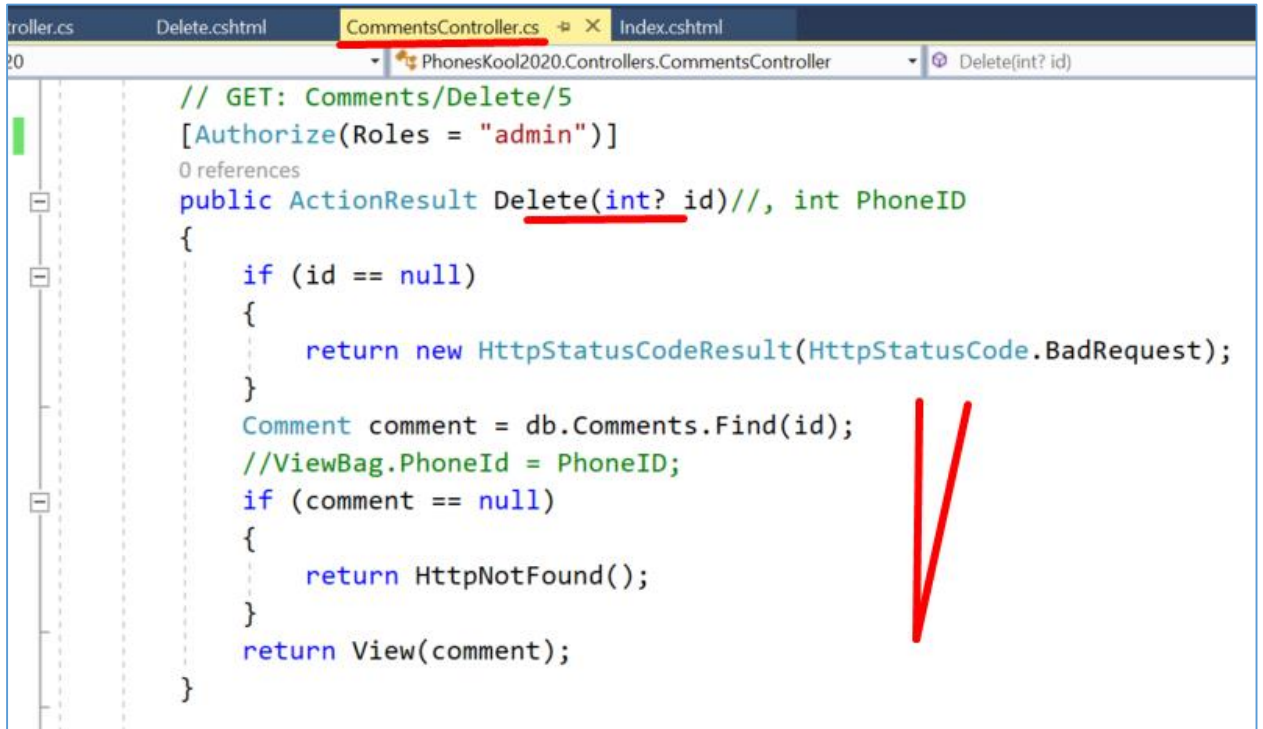
A red checkmark is placed next to the `<link type="text/css" rel="stylesheet" href="~/Content/PhonesStyles.css" />` line, indicating that the styles are correctly linked.

Удаление комментариев – роль админ

Удаление комментариев имеет право только админ. Для удаления необходимо знать id комментария и id продукта для возврата на страницу продукт Details

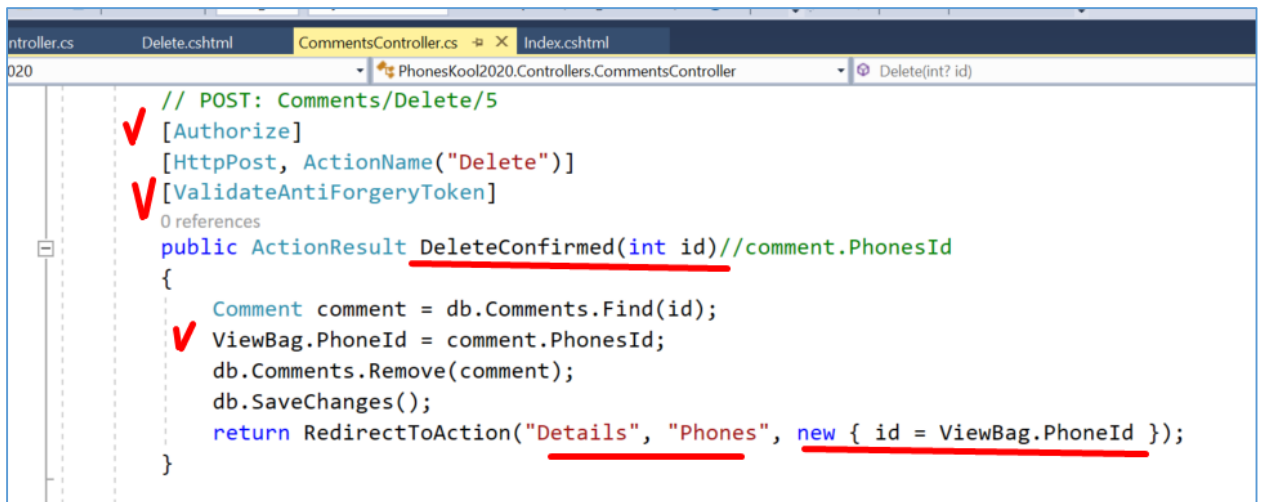
Изменяем код **Controller/CommentsController**

Метод **Delete**



```
// GET: Comments/Delete/5
[Authorize(Roles = "admin")]
public ActionResult Delete(int? id), int PhoneID
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Comment comment = db.Comments.Find(id);
    //ViewBag.PhoneId = PhoneID;
    if (comment == null)
    {
        return HttpNotFound();
    }
    return View(comment);
}
```

Возврат **для обработки формы Delete**



```
// POST: Comments/Delete/5
[Authorize]
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)//comment.PhonesId
{
    Comment comment = db.Comments.Find(id);
    ViewBag.PhoneId = comment.PhonesId;
    db.Comments.Remove(comment);
    db.SaveChanges();
    return RedirectToAction("Details", "Phones", new { id = ViewBag.PhoneId });
}
```

Файл Views/Comments/Delete.cshtml

```
1 @model PhonesKool2020.Models.Comment
2 @using Microsoft.AspNet.Identity;
3
4 @{
5     ViewBag.Title = "Delete";
6 }
7 <h2>Delete</h2>
8
9 @if (Model.UserName == User.Identity.GetUserName() || User.Identity.GetUserName() == "admin")
10 {
11     <h3>Are you sure you want to delete this?</h3>
12     <div>
13         <h4>Comment</h4>
14         <hr />
15         <dl class="dl-horizontal">
16             <dt>
17                 @Html.DisplayNameFor(model => model.UserName)
18             </dt>
19             <dd>
20                 @Html.DisplayFor(model => model.UserName)
21             </dd>
22             <dt>
23                 @Html.DisplayNameFor(model => model.Comments)
24             </dt>
```

Начало кода

```
25         <dd>
26             @Html.DisplayFor(model => model.Comments)
27         </dd>
28         <dt>
29             @Html.DisplayNameFor(model => model.DatePublish)
30         </dt>
31         <dd>
32             @Html.DisplayFor(model => model.DatePublish)
33         </dd>
34         <dt>
35             @Html.DisplayNameFor(model => model.Phone.Name)
36         </dt>
37         <dd>
38             @Html.DisplayFor(model => model.Phone.Name)
39         </dd>
40     </dl>
41     @using (Html.BeginForm())
42     {
43         @Html.AntiForgeryToken()
44         <p>
45             <input type="submit" value="Delete" /> |
46             @Html.ActionLink("Back to Phone", "Details", "Phones", new { id = Model.PhonesId }, null)
47         </p>
48     }
49 </div>
50 }
51 else
52 {
53     <h2>У вас нет прав удалять комментарий!</h2>
54     <h3>Phone Title: @Html.DisplayFor(model => model.Phone.Name)</h3>
55     @Html.ActionLink("Back to Phone", "Details", "Phones", new { id = ViewBag.PhoneId }, null)
56 }
```

Продолжение кода

Задание: Протестируйте работу проекта для разных ролей