**www.eyeglasses123.com**

**\*\* Add one more filter in Facet Navigation \*\***

I am glad to made this facet navigation. Which is very faster as compare to all our facet navigation. It is fully dynamic and very easy to add one more filter in it. So let’s start to learn how to add one more filter in ‘Facet Navigation’.

In this document I take example of to add one more filter in Brand Product List page.

First of all welcome to learn how to ‘Add one more filter in Facet Navigation’. So let’s start with step by step.

**Step-1 : Required files**

There are 2 files required to make facet navigation in **Brand Product List** page which are shown at below with its location.

* \EyeGlassFront\EyeGlasses\Controllers\BrandController.cs
* \EyeGlassFront\EyeGlasses\Views\Brand\Brand.cshtml

**Step-2 : Start with example to add new filter ‘Frame Color’**

For better understand I am add one more filter which is **FrameColor.**

I am started from its Controller which is **BrandController.cs**

**Step-3 : Required data from data base.**

There are 3 types of data required which are listed below

1. Filter Master Data
2. Filter Mapping Data
3. Filter Product Data

A data come from below Stored Procedures

1. GuiGetBrandFacetedMasterEG
2. GuiGetBrandAllSearchdataEG
3. GuiGetBrandProductsEG
4. **SP: GuiGetBrandFacetedMasterEG**

Go to end of S.P. **GuiGetBrandFacetedMasterEG** and add below query which is get all frame color data from it’s master table. It is also illustrated in below screenshot.

-- start frame color Master

SELECT DISTINCT ISNULL(fc.FrameStandardColorName,'') AS 'Name', ISNULL(fc.FrameStandardColorCode,'') AS 'ColorCode', ISNULL(fc.ImageName,'') AS 'ImageName', ISNULL(fc.DisplayOrder,0) AS 'DisplayOrder', fc.FrameStandardColorID

FROM tb\_FrameStandardColor\_MST fc

INNER JOIN tb\_FrameStandardColorMapping fcm on fcm.FrameStandardColorID = fc.FrameStandardColorID

INNER JOIN tb\_Product P on p.ProductID = fcm.ProductID

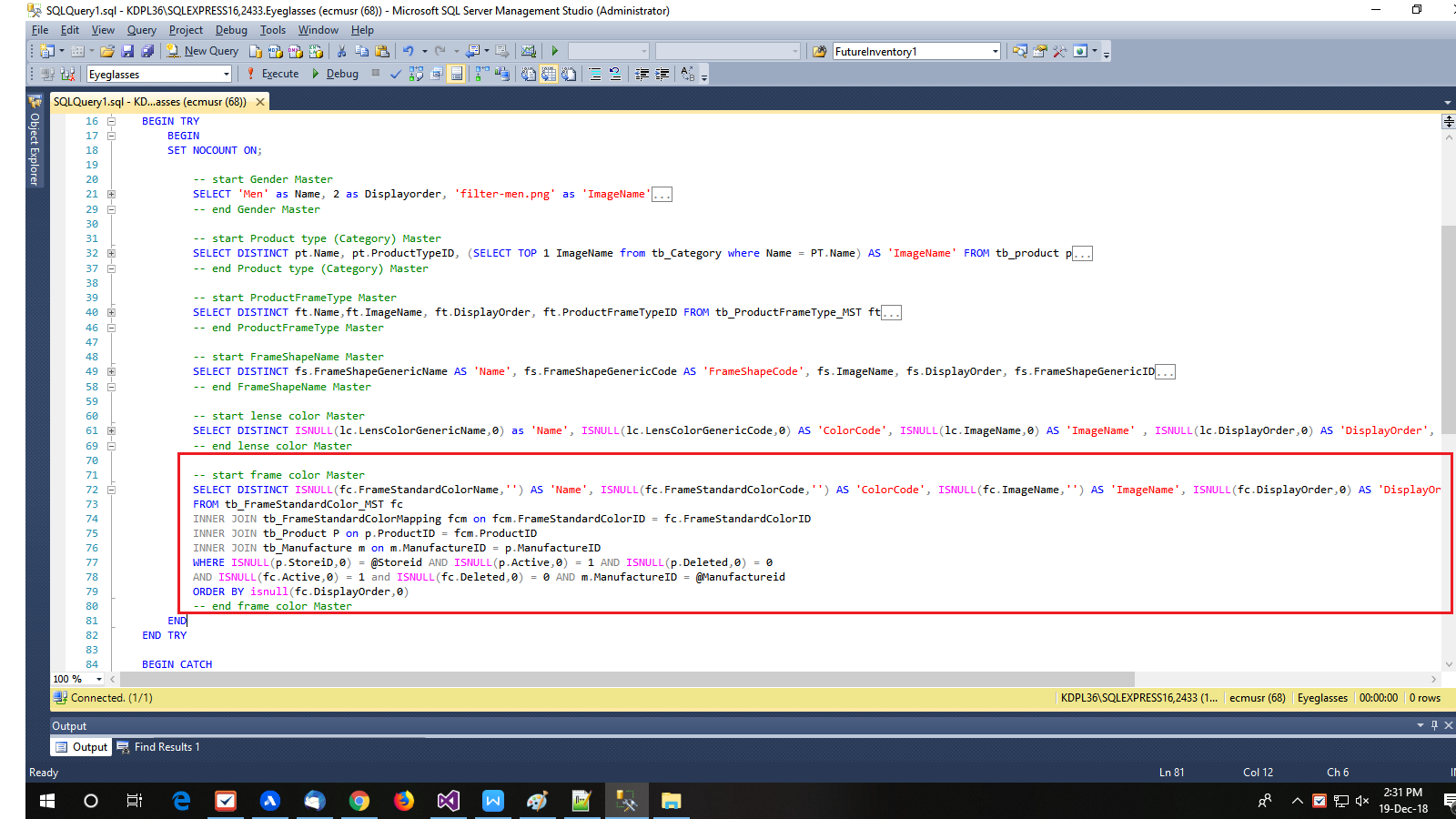
INNER JOIN tb\_Manufacture m on m.ManufactureID = p.ManufactureID

WHERE ISNULL(p.StoreiD,0) = @Storeid AND ISNULL(p.Active,0) = 1 AND ISNULL(p.Deleted,0) = 0

AND ISNULL(fc.Active,0) = 1 and ISNULL(fc.Deleted,0) = 0 AND m.ManufactureID = @Manufactureid

ORDER BY isnull(fc.DisplayOrder,0)

-- end frame color Master



1. **SP: GuiGetBrandAllSearchdataEG**

Go to the end of the S.P. **GuiGetBrandAllSearchdataEG** and add below query which is get all mapping data of frame color from it’s mapping table.

-- start select frame color

SELECT DISTINCT fc.FrameStandardColorName AS 'Name',p.ProductID,fc.FrameStandardColorID AS ID, case When isnull(p.salePrice,0) > 0 then isnull(p.salePrice,0) else isnull(p.price,0) end price, isnull(fc.DisplayOrder,0) AS DisplayOrder

FROM tb\_product p

INNER JOIN tb\_ProductInfo pin ON PIN.ProductID = p.ProductID

INNER JOIN tb\_FrameStandardColorMapping fcm ON fcm.ProductID = pin.ProductID

INNER JOIN tb\_FrameStandardColor\_MST fc ON fc.FrameStandardColorID= fcm.FrameStandardColorID

INNER JOIN tb\_Manufacture m on m.ManufactureID = p.ManufactureID

INNER JOIN tb\_ProductCategory ON p.ProductID = tb\_ProductCategory.ProductID

INNER JOIN tb\_Category c ON tb\_ProductCategory.CategoryID = c.CategoryID

WHERE ISNULL(c.Deleted,0)=0 AND ISNULL(c.Active,0)=1 AND ISNULL(c.StoreID,0)= @StoreID

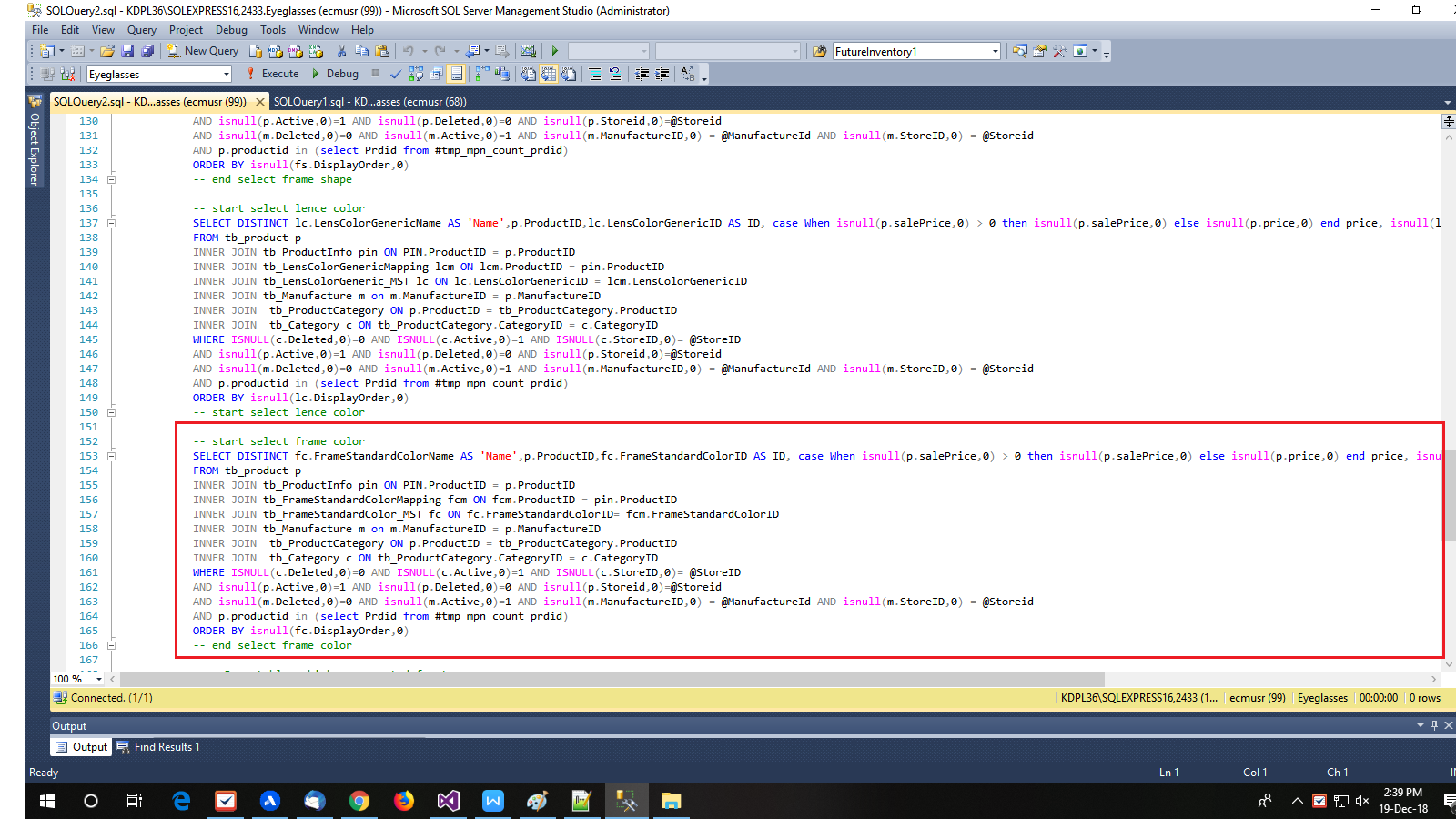
AND isnull(p.Active,0)=1 AND isnull(p.Deleted,0)=0 AND isnull(p.Storeid,0)=@Storeid

AND isnull(m.Deleted,0)=0 AND isnull(m.Active,0)=1 AND isnull(m.ManufactureID,0) = @ManufactureId AND isnull(m.StoreID,0) = @Storeid

AND p.productid in (select Prdid from #tmp\_mpn\_count\_prdid)

ORDER BY isnull(fc.DisplayOrder,0)

-- end select frame color

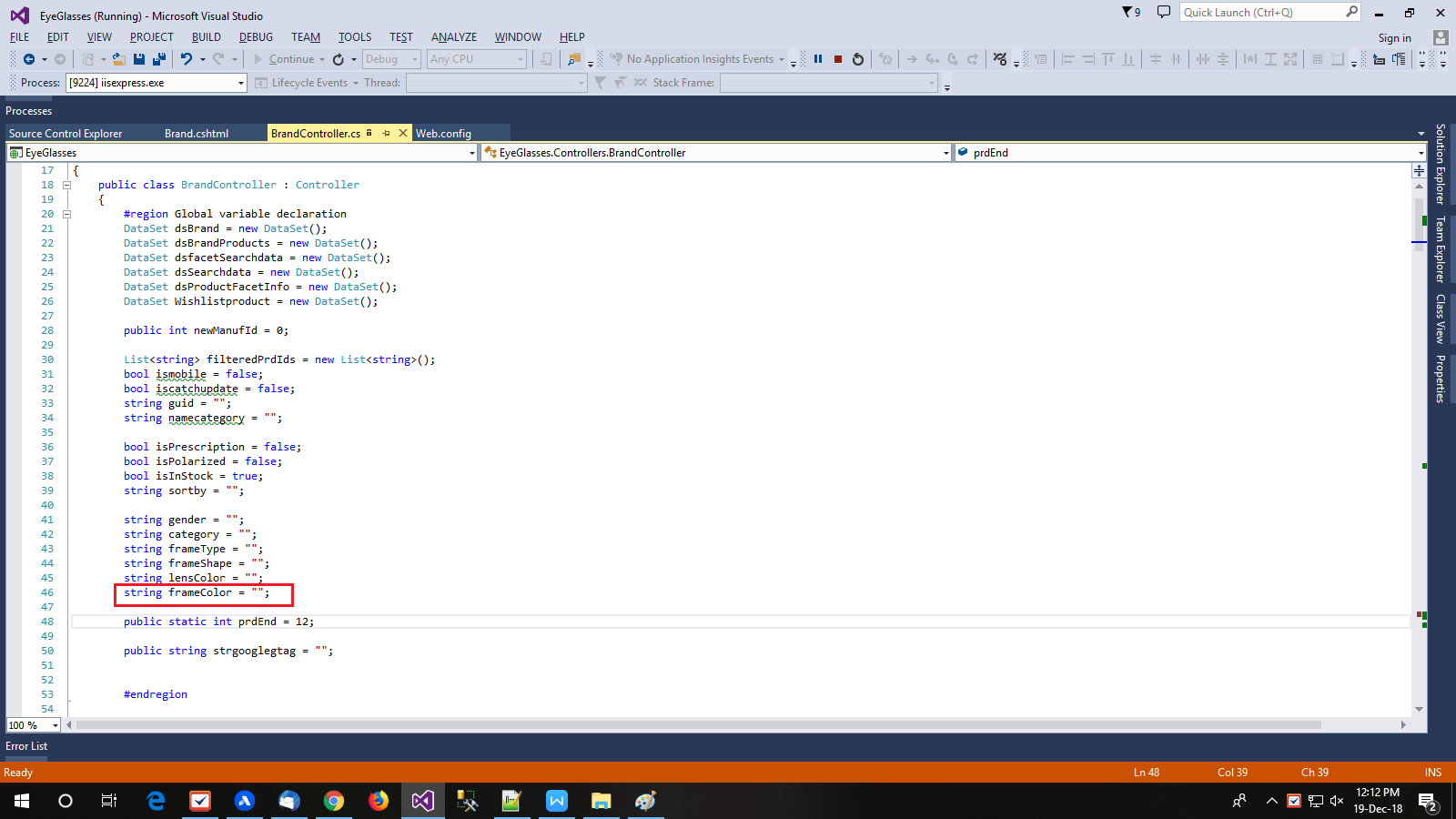


1. **SP: GuiGetBrandProductsEG**

Nothing to add in this StoredProcedure whenever add new filter.

**Step-4 : Changes in Controller page**

First of all declare a global variable for new filter.



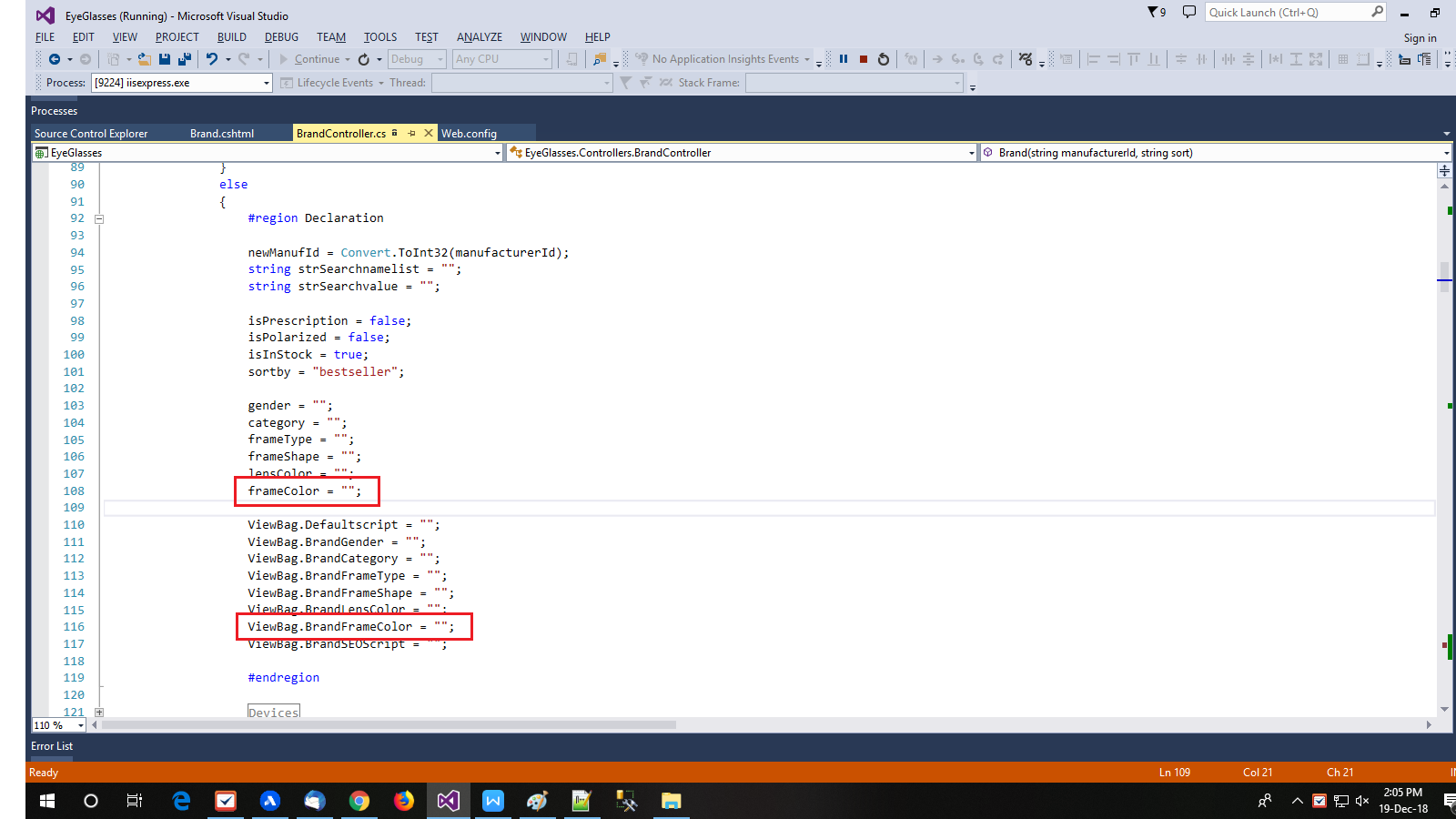
Add below code snippet as per shown in screenshot.

string frameColor = "";

Now go to in Brand Constructor which is shown below.

public ActionResult Brand(string manufacturerId, string sort){…}

**Step-5 : Clear old filter value and ViewBag data of FrameColor.**

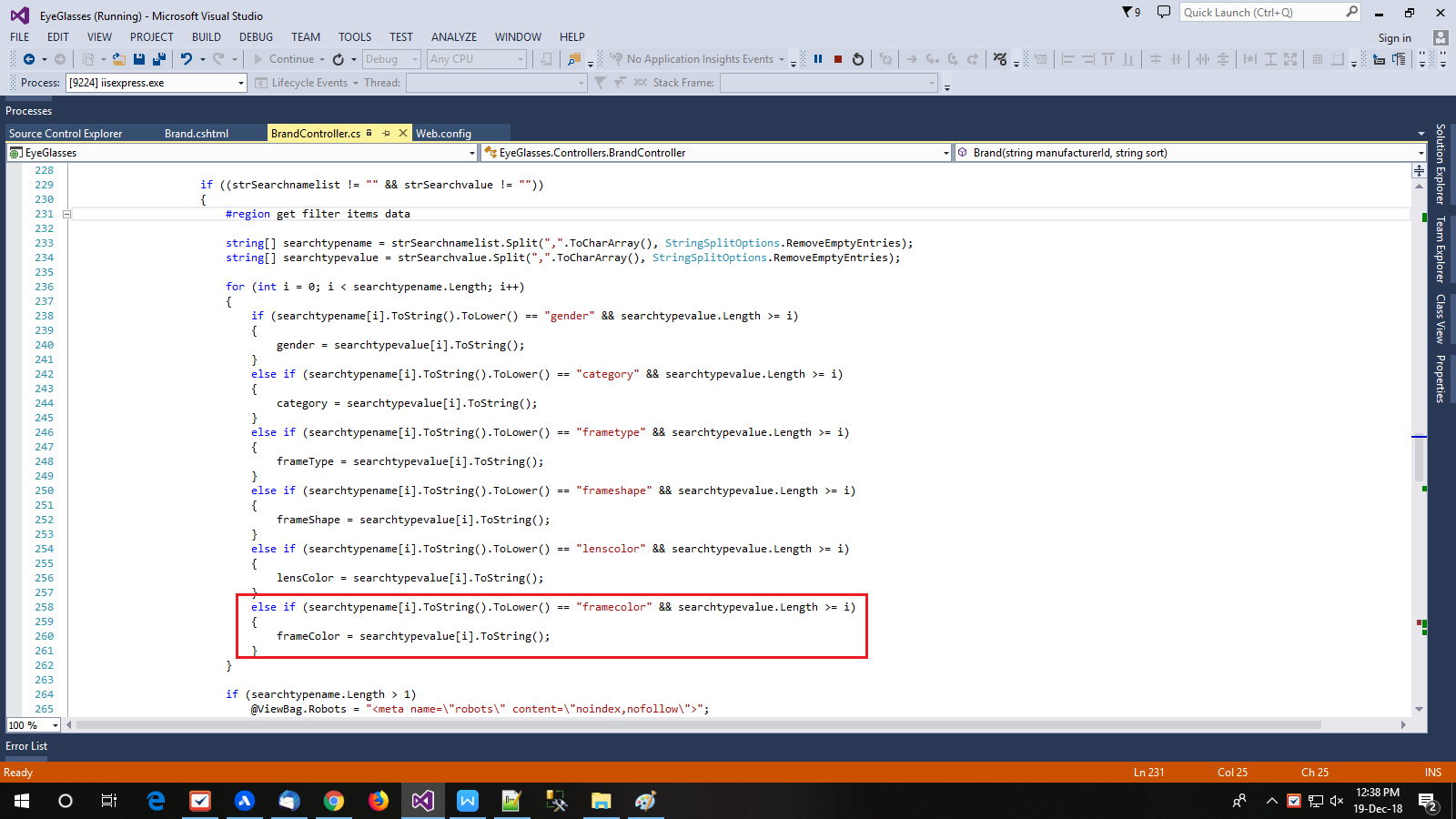


Add below code snippet as per shown in above screenshot.

frameColor = "";

ViewBag.BrandFrameColor = "";

**Step-6 : Get filter items from URL to global variables**



Add below code snippet as per shown in above screenshot.

**e**lse if (searchtypename[i].ToString().ToLower() == "framecolor" && searchtypevalue.Length >= i)

{

frameColor = searchtypevalue[i].ToString();

}

**Step-7 : Add StringBuilder to store its HTML design and pass filer value to function BindFacetData.**

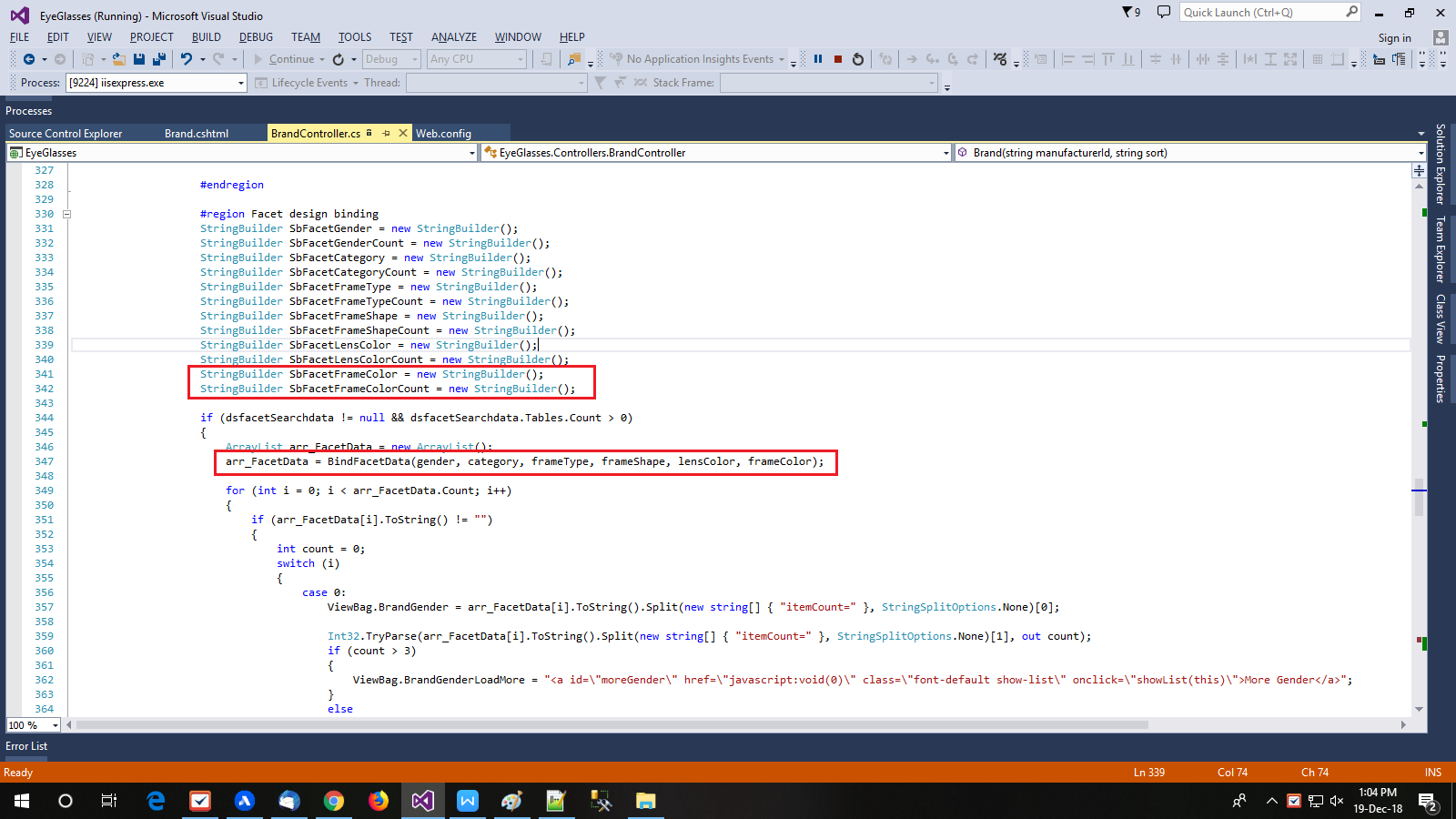
Add below code snippet as per shown in below screenshot.

StringBuilder SbFacetFrameColor = new StringBuilder();

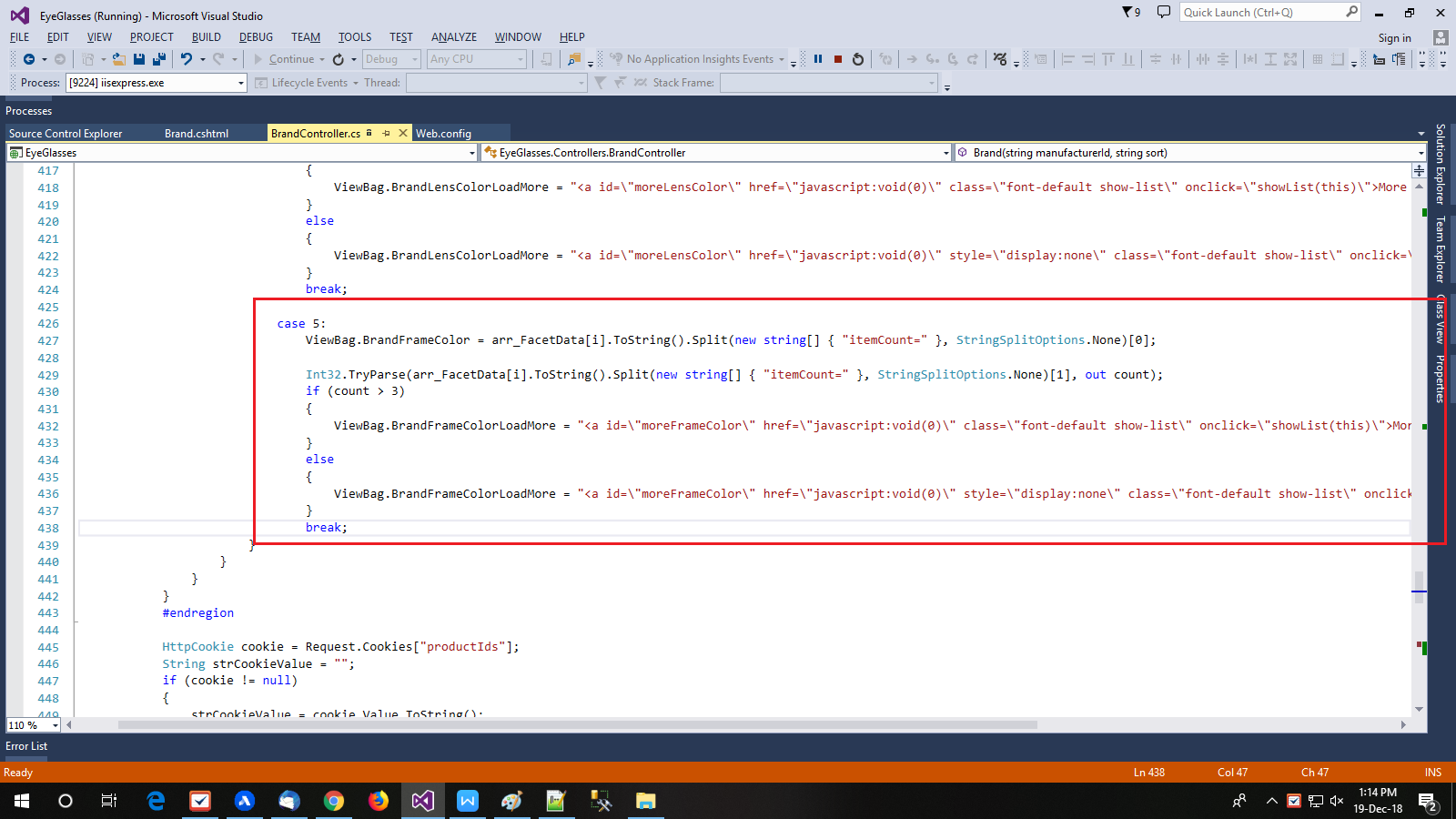
StringBuilder SbFacetFrameColorCount = new StringBuilder();

Add filter variable add last in function BindFacetData. Lets do it.

arr\_FacetData = BindFacetData(gender, category, frameType, frameShape, lensColor, frameColor);



**Step-8 : Now get HTML design string from arr\_FacetData and store it on ViewBage**



Now add new case for a filter **FrameColor** which is shown below.

case 5:

ViewBag.BrandFrameColor= arr\_FacetData[i].ToString().Split(new string[] { "itemCount=" }, StringSplitOptions.None)[0];

Int32.TryParse(arr\_FacetData[i].ToString().Split(new string[] { "itemCount=" }, StringSplitOptions.None)[1], out count);

if (count > 3)

{

ViewBag.BrandFrameColorLoadMore= "<a id=\"moreFrameColor\" href=\"javascript:void(0)\" class=\"font-default show-list\" onclick=\"showList(this)\">More Colors</a>";

}

else

{

ViewBag.BrandFrameColorLoadMore= "<a id=\"moreFrameColor\" href=\"javascript:void(0)\" style=\"display:none\" class=\"font-default show-list\" onclick=\"showList(this)\">More Colors</a>";

}

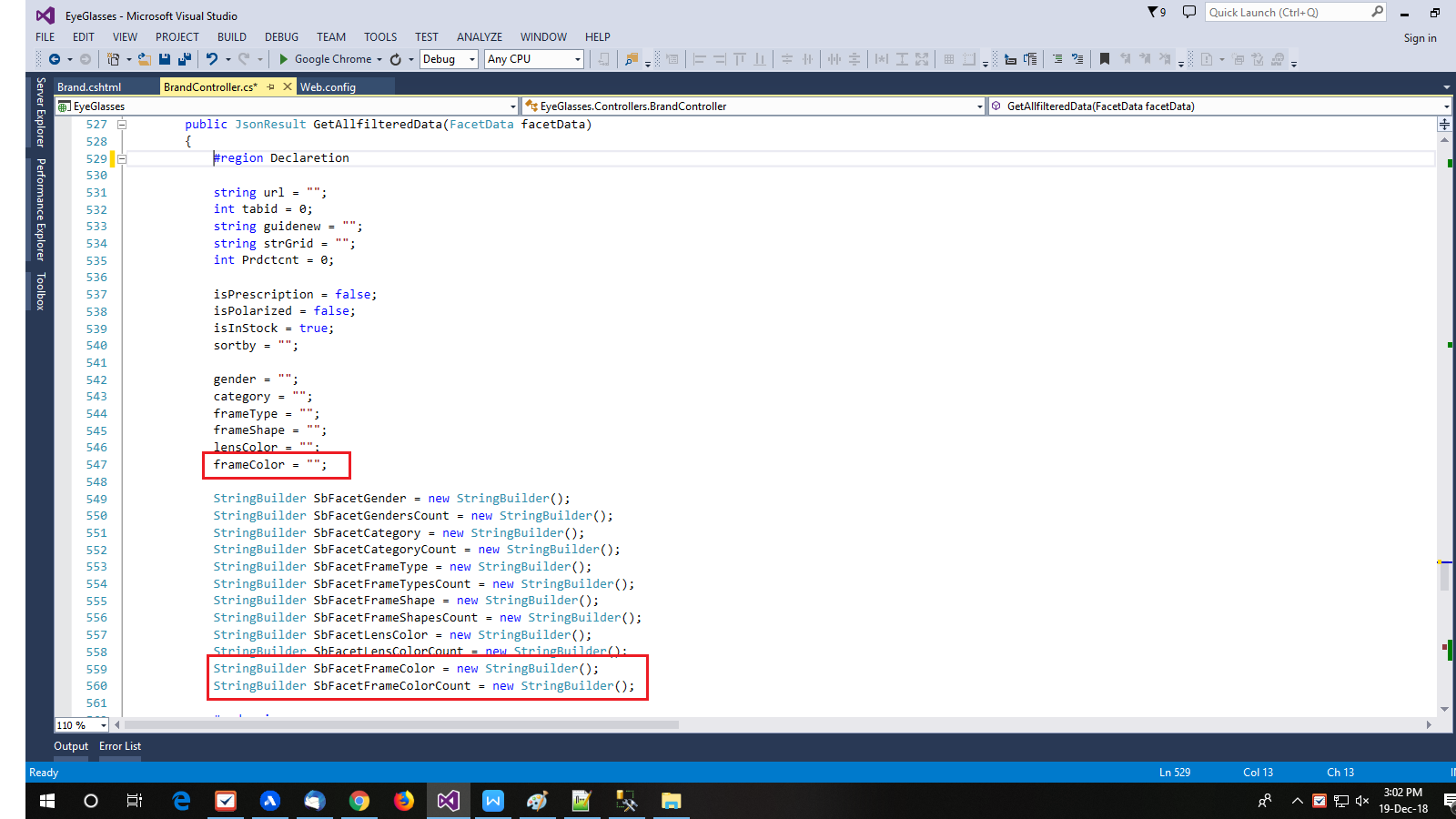
break;

**Step-9 : Go to function** **GetAllfilteredData()**

public JsonResult GetAllfilteredData(FacetData facetData){……}

This function is call when user select any filter item from left side menu.

Clear filter and declare StringBuilder for it.



Add below code snippet as per shown in above screen.

frameColor = "";

StringBuilder SbFacetFrameColorCount = new StringBuilder();

**Step-10 : Get filters from model named ‘facetData’ and store it in global variables**

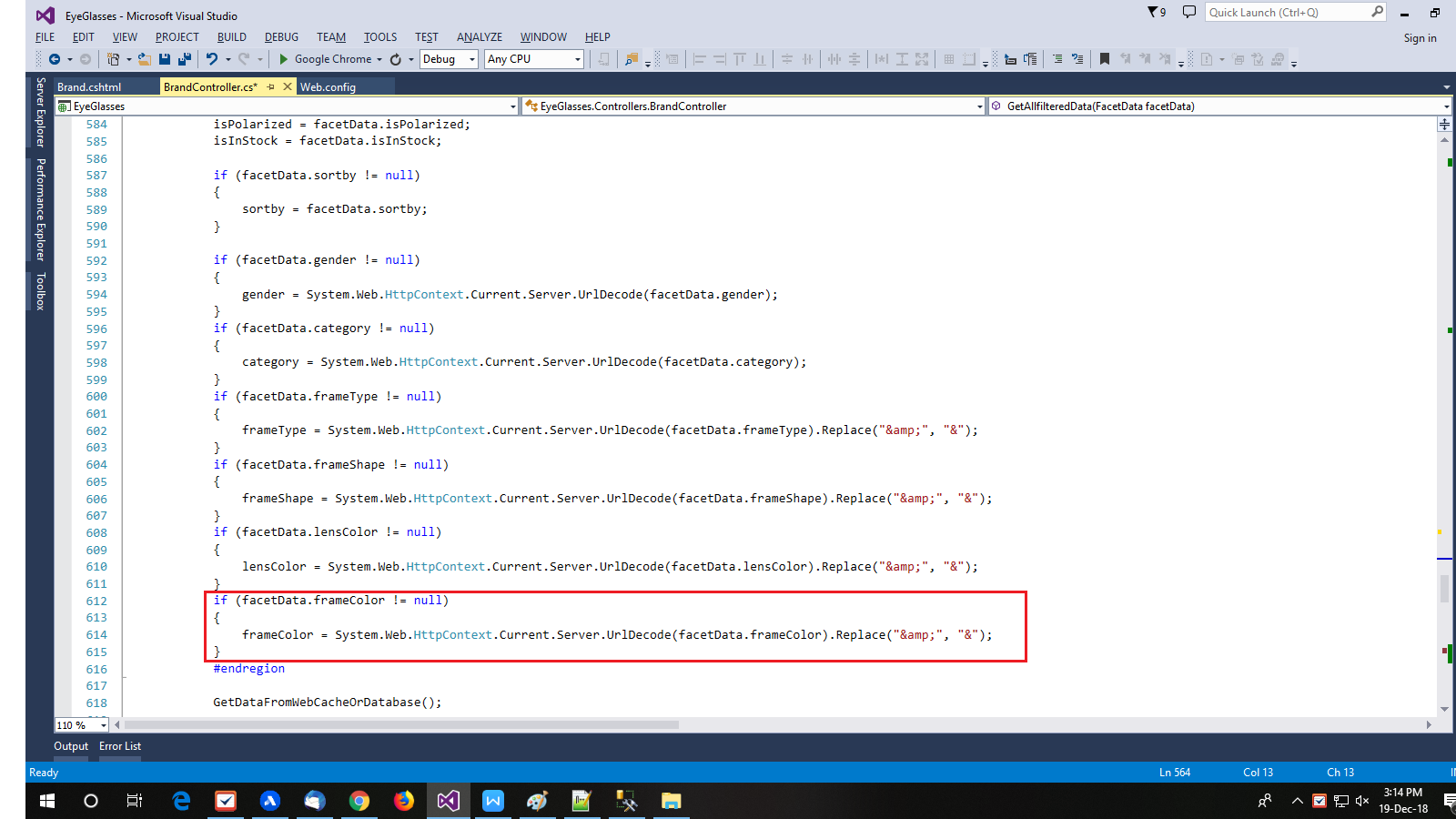
Add below code snippet to get filter items data from model named facetData and store it in global variables. Also it’s illustrated in below screenshot.

if (facetData.frameColor != null)

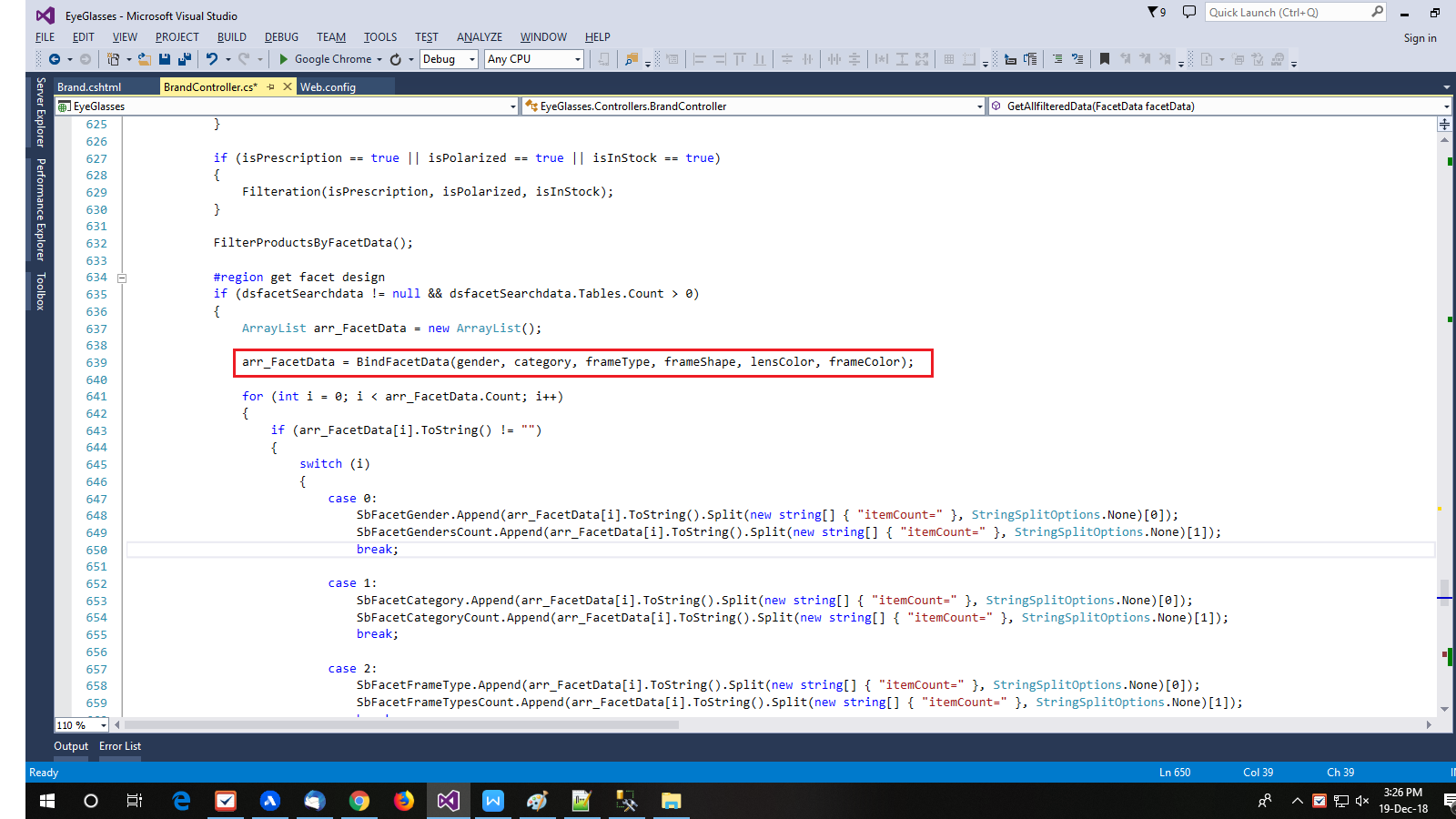
{

frameColor = System.Web.HttpContext.Current.Server.UrlDecode(facetData.frameColor).Repla ce("&amp;", "&");

}



**Step-11 : Pass filter data to get new facet design as per filters and return in JSON format.**



Add one more parameter **frameColor** to end of the function **BindFacetData** as below line and also shown in above screenshot.

**arr\_FacetData = BindFacetData(gender, category, frameType, frameShape, lensColor, frameColor);**

This function returns an array of filters HTML design and store it in **arr\_FacetData**.

Now get HTML design from array and store it in StringBuilder which is above declared with name **SbFacetFrameColorCount** in step-9**.**

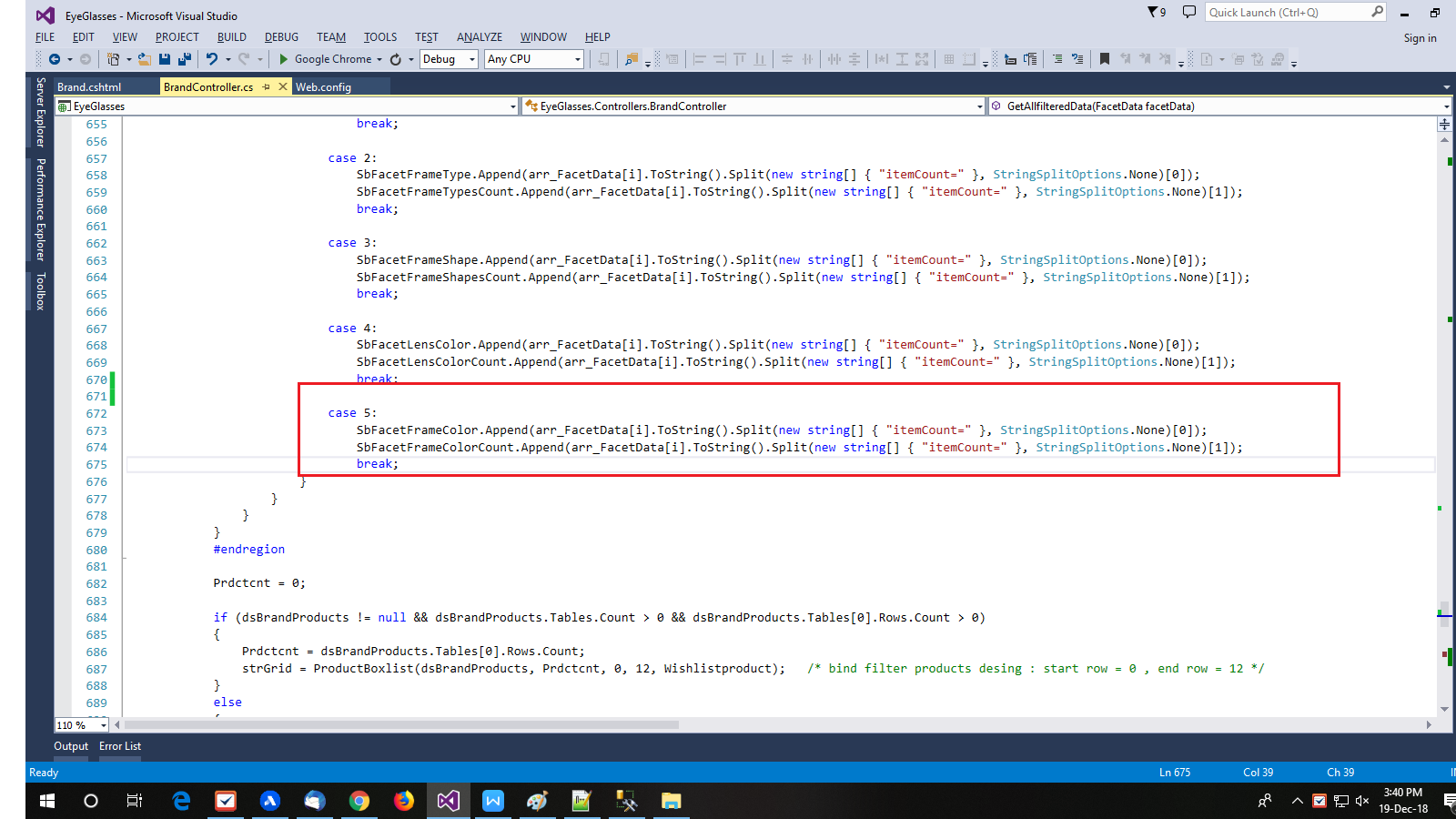
**Add a new case for Frame Color as per bellow code snippet**

case 5: SbFacetFrameColor.Append(arr\_FacetData[i].ToString().Split(new string[] { "itemCount=" }, StringSplitOptions.None)[0]);

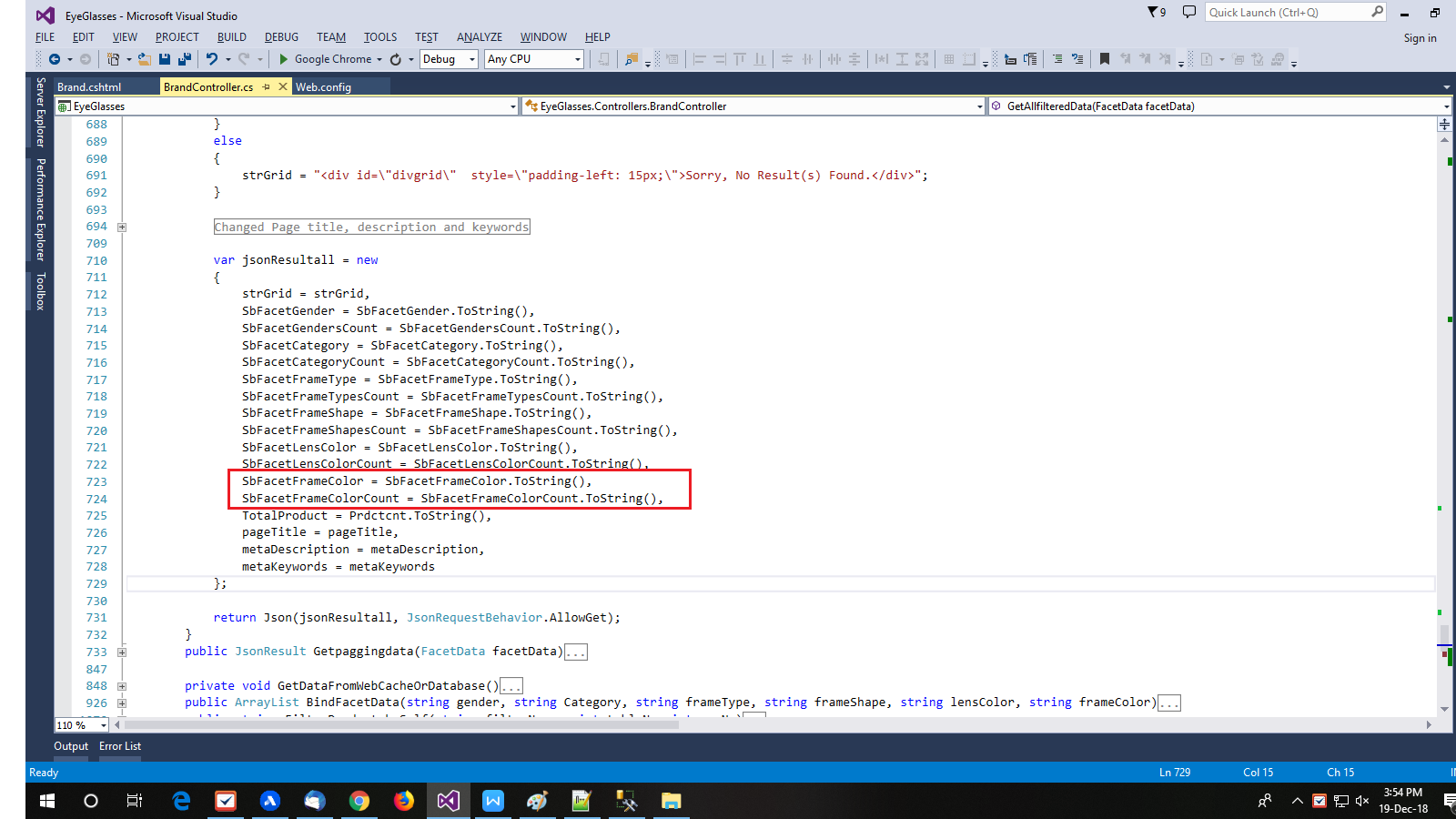
SbFacetFrameColorCount.Append(arr\_FacetData[i].ToString().Split(new string[] { "itemCount=" }, StringSplitOptions.None)[1]);

break;

So now code look like below.



**Return new filter HTML design and filter items count in JSON format.**



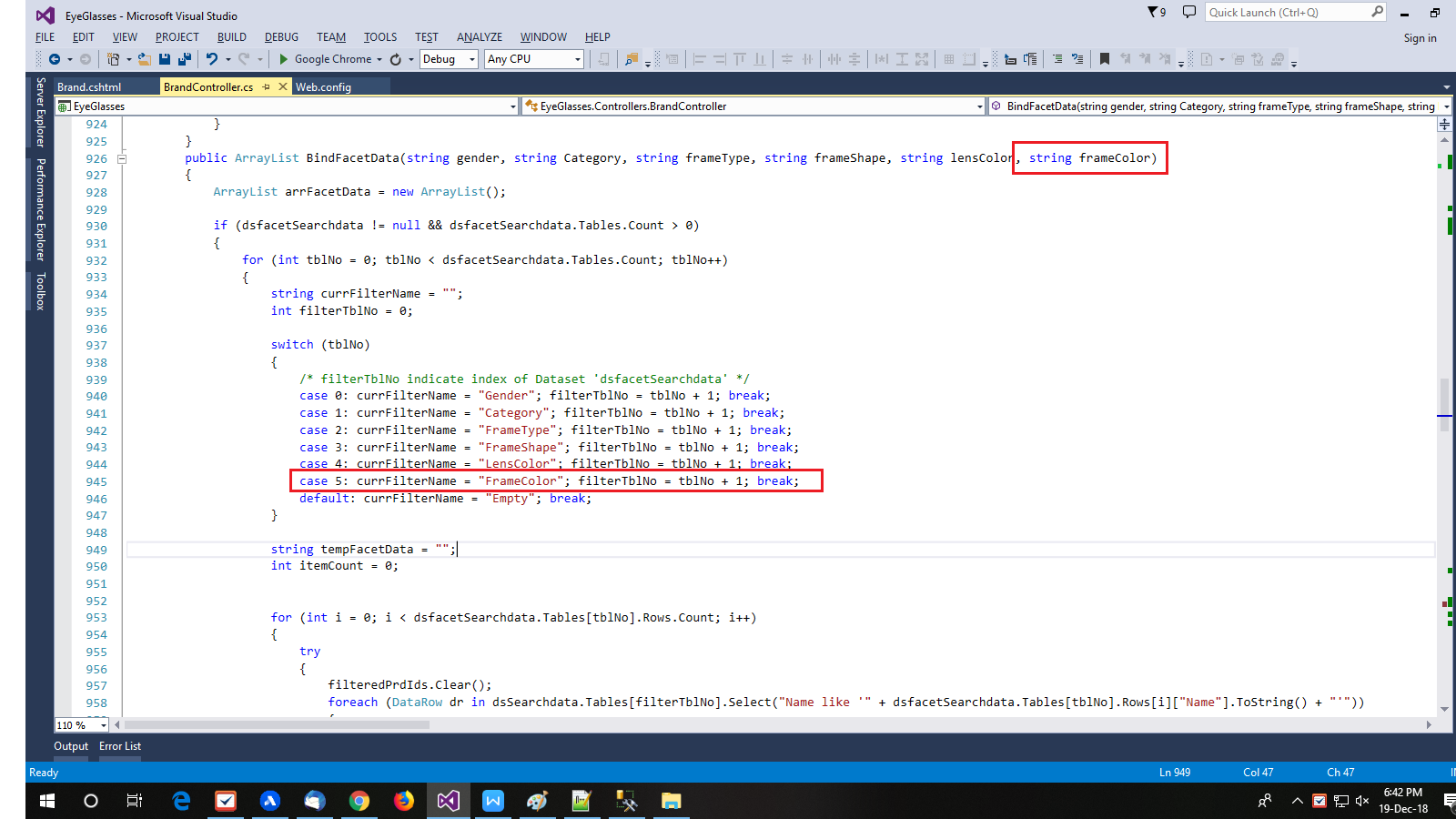
Add below code snippet as per shown in above screenshot.

**SbFacetFrameColor = SbFacetFrameColor.ToString(),**

**SbFacetFrameColorCount = SbFacetFrameColorCount.ToString(),**

**Step-12 : Go to in function BindFacetData**

This function returns an HTML design of filters as per applied filters.



Add below snippet code to add one more parameter **frameColor** and add one more case in SwitchCase which are also shown in above screenshot.

public ArrayList BindFacetData(string gender, string Category, string frameType, string frameShape, string lensColor, string frameColor){…}

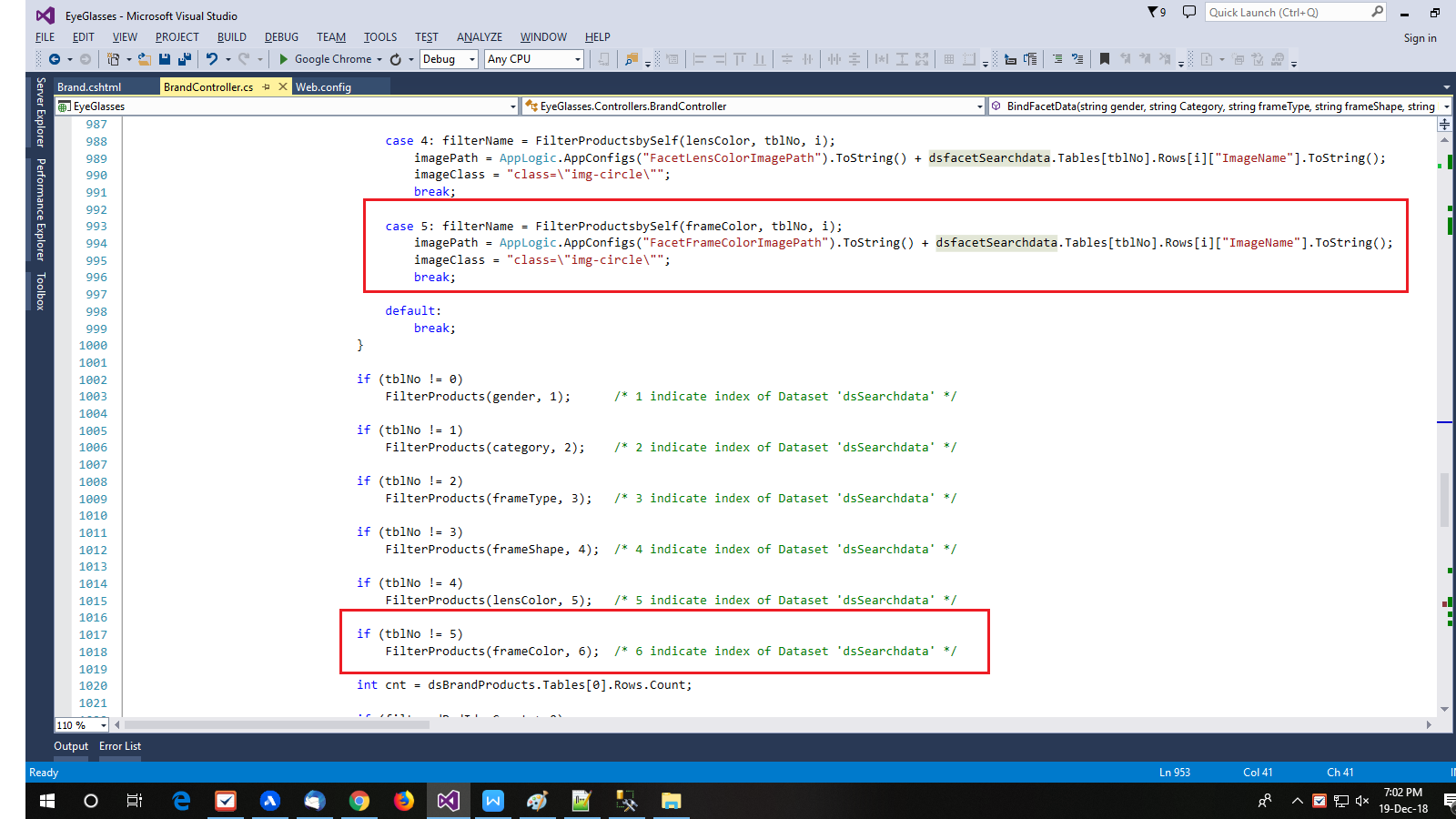
case 5: currFilterName = "FrameColor"; filterTblNo = tblNo + 1; break;

**Note:** Make sure that here **tblNo + 1** is index of DataTable in DataSet for FrameColor.

Here dataset contains MasterData of filters, So must check index of filter table in dataset is match with **tblNo + 1**.

Now go into below for loop,

**for (int i = 0; i < dsfacetSearchdata.Tables[tblNo].Rows.Count; i++)**



And add below snippet code as shown in above screenshot.

case 5: filterName = FilterProductsbySelf(frameColor, tblNo, i);

imagePath = AppLogic.AppConfigs("FacetFrameColorImagePath").ToString() + dsfacetSearchdata.Tables[tblNo].Rows[i]["ImageName"].ToString();

imageClass = "class=\"img-circle\"";

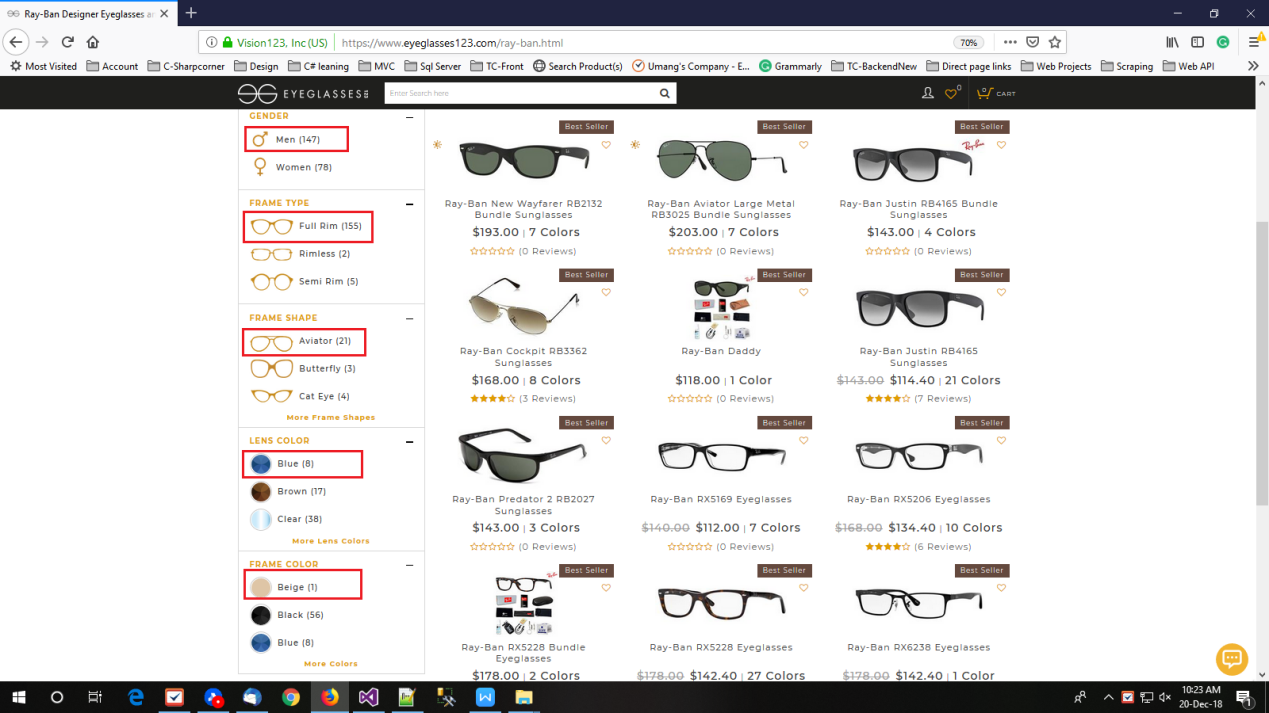
break;

if (tblNo != 5)

FilterProducts(frameColor, 6); /\* 6 indicate index of Dataset 'dsSearchdata' \*/

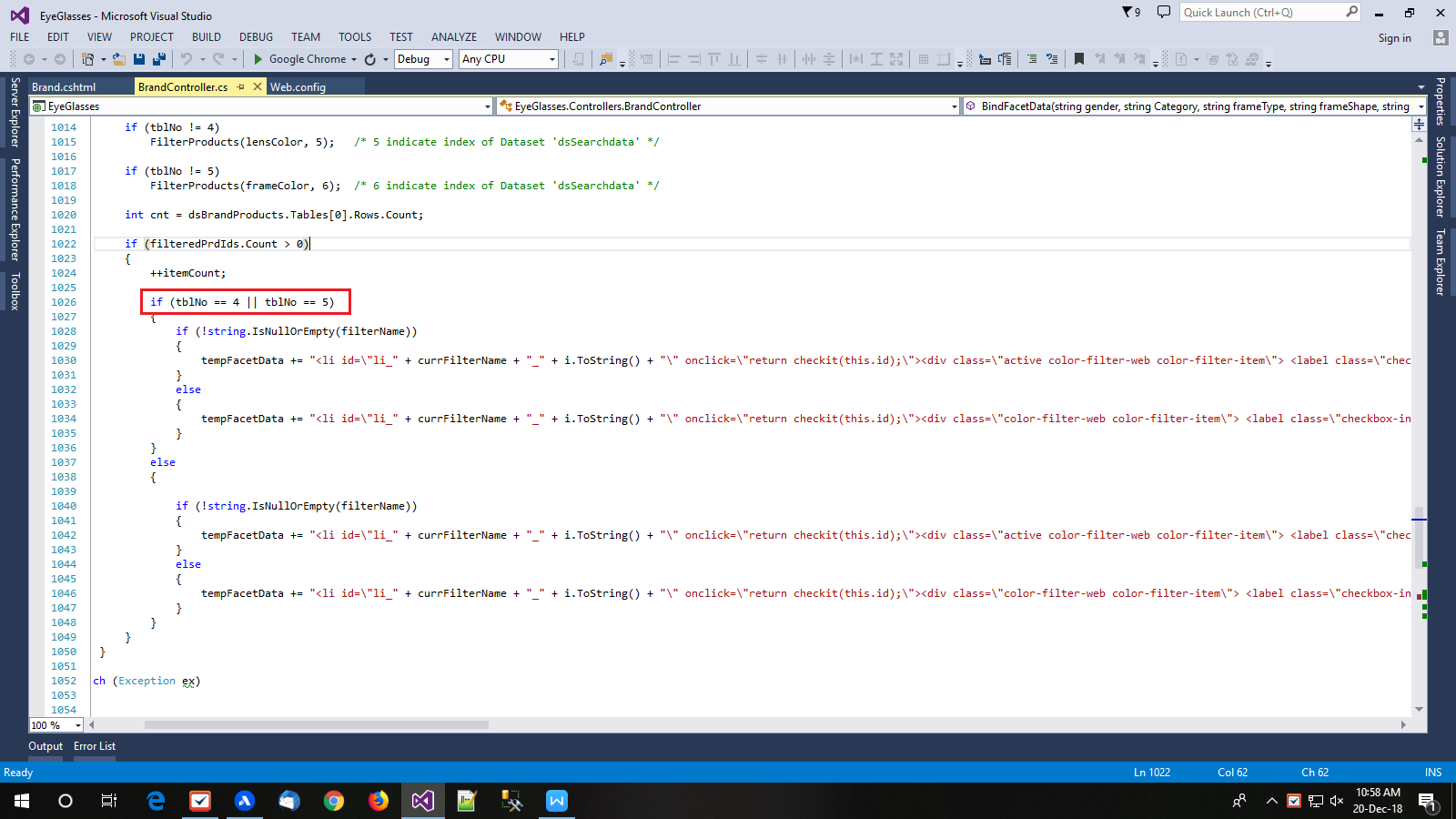
There are two type of design to bind filters items

1. Design for color filters
2. Common design for all remaining filters



You can see that common design for color is ‘Lens Color’ and ‘Frame Color’. And common design for remaining filters which are ‘Gender’, ‘Frame Type’, and ‘Frame Shape’.

Now add Frame Color filter items design, Illustrate below screenshot to understand how to the add it.



In above screenshot **tblNo == 4** ||**tblNo == 5** are indicate that common design for color. **tblNo == 4** is for ‘Lens Color’ and **tblNo == 5** is for ‘Frame Color’. And common design for all remaining filter item is in else part.

**Step-13 : Go to in function ‘FilterProductsByFacetData’**

This function filter a products from DataSet as per filter items and store filtered result on same DataSet which is **dsBrandProducts**.

Now first of all check filter is not empty like below, so add code **|| frameColor != ""**

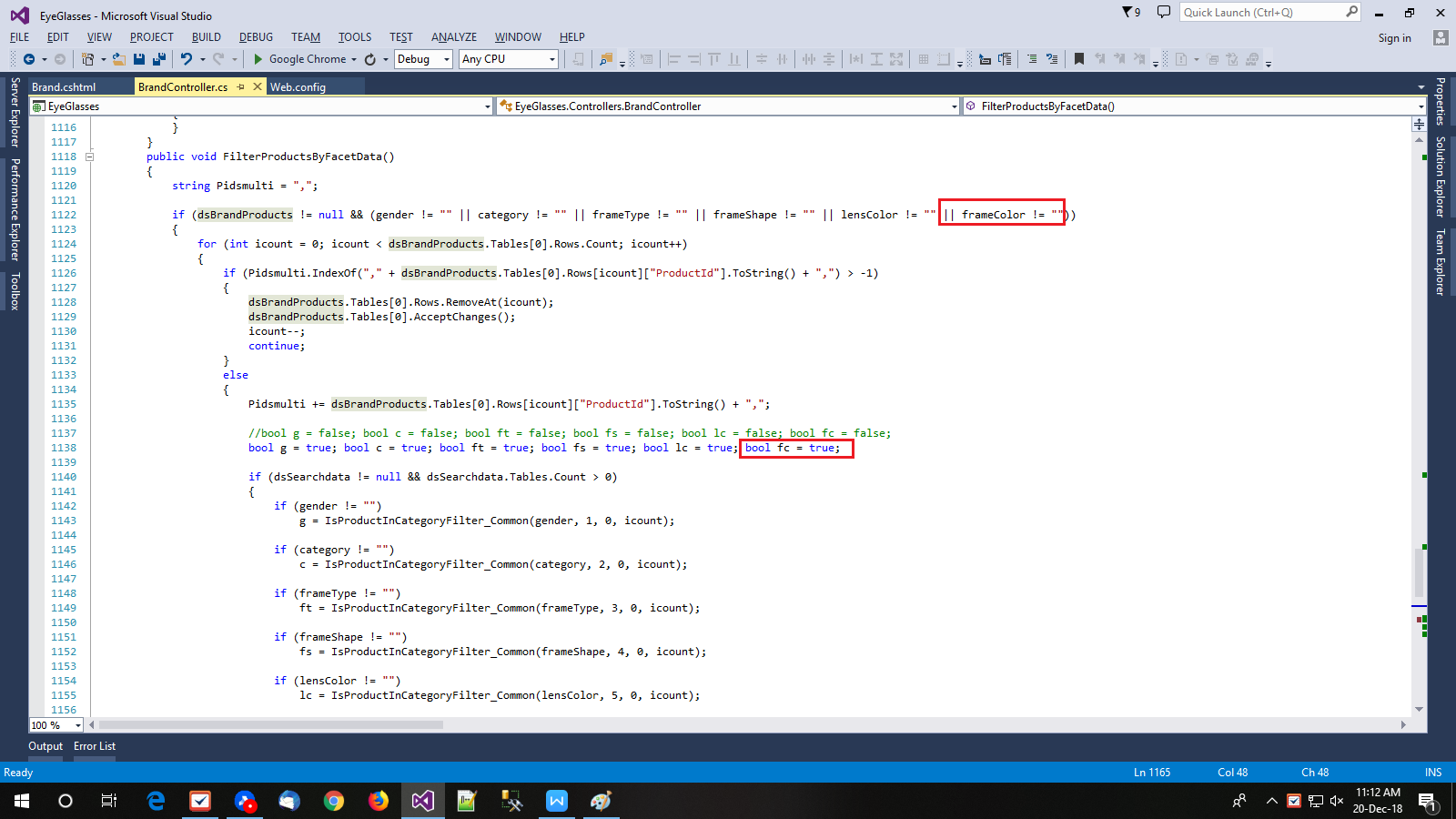
at the end of if condition so it look like below.

if (dsBrandProducts != null && (gender != "" || category != "" || frameType != "" || frameShape != "" || lensColor != "" **|| frameColor != ""**))

Also you need to declare Boolean variable to check product is exist or not for this filter. So it look like below and also illustrate in below screenshot.

bool g = true; bool c = true; bool ft = true; bool fs = true; bool lc = true;

**bool fc = true;**

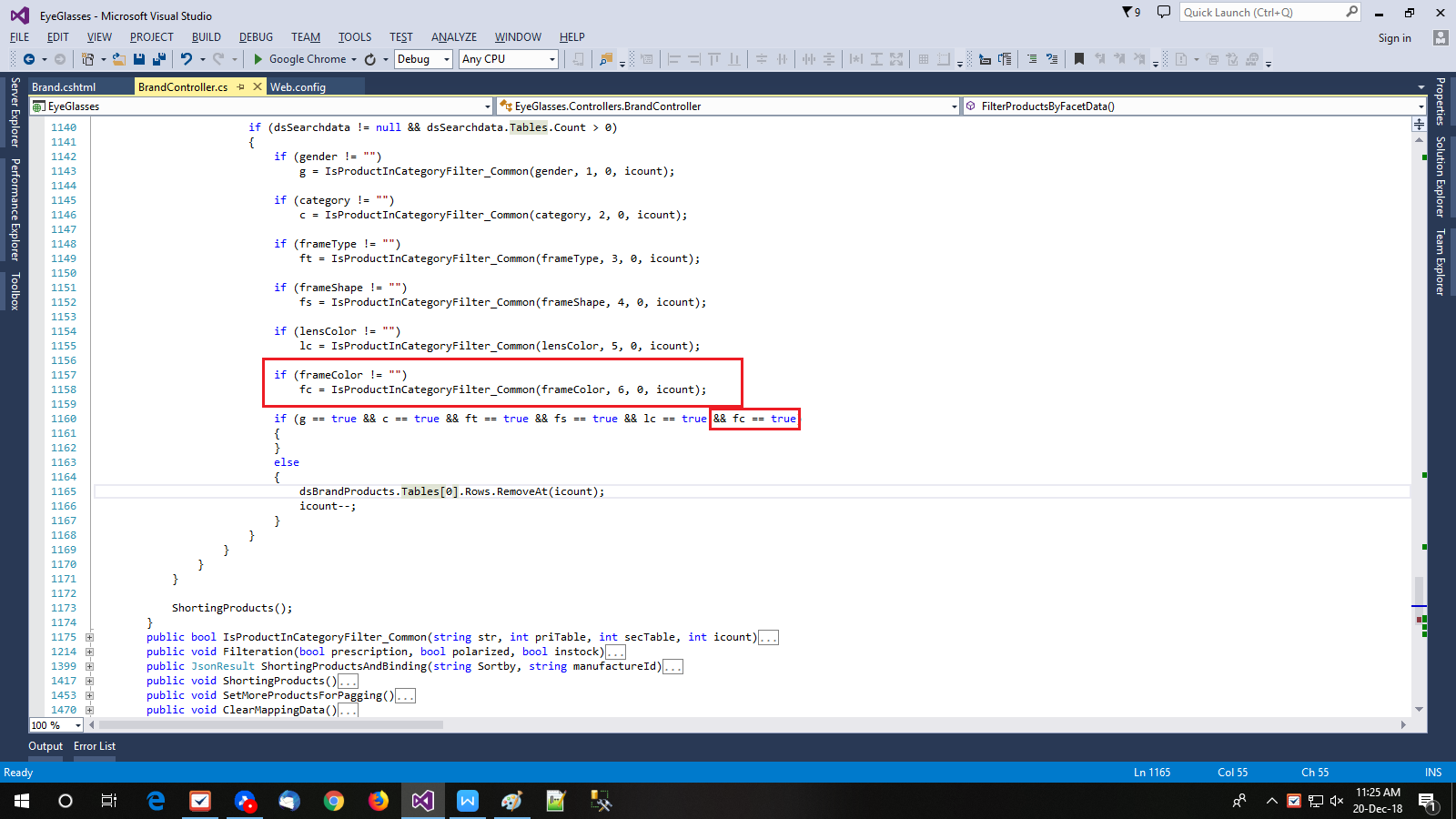


Now check if filter item is not empty then check that current product is in this filter. Add below code snippet for it.

**if (frameColor != "")**

**fc = IsProductInCategoryFilter\_Common(frameColor, 6, 0, icount);**

And check **fc** is true or not, if it is true then do nothing other vise product is remove from DataSet. Add **&& fc == true** to the end of is condition which illustrate in below screenshot.



So all changes in controller side is done. Now we are go to the view of Brand Product List page.

**Step-14 : Changes in View page**

File name and it’s path in our project is shown below.

\EyeGlassFront\EyeGlasses\Views\Brand\Brand.cshtml

Now add design code which is come from controller. Add below code, so it look like following screenshot.

@{

if (ViewBag.BrandFrameColor != "")

{

<div class="side-box-sec" id="divFrameColorFacet">

<div class="side-box-sec-title">Frame Color</div>

<div class="side-box-sec-desc">

<form>

<ul id="ulFrameColor" class="list-unstyled limited-list filter-content filter-box-3">

@Html.Raw(ViewBag.BrandFrameColor)

</ul> @Html.Raw(ViewBag.BrandFrameColorLoadMore)

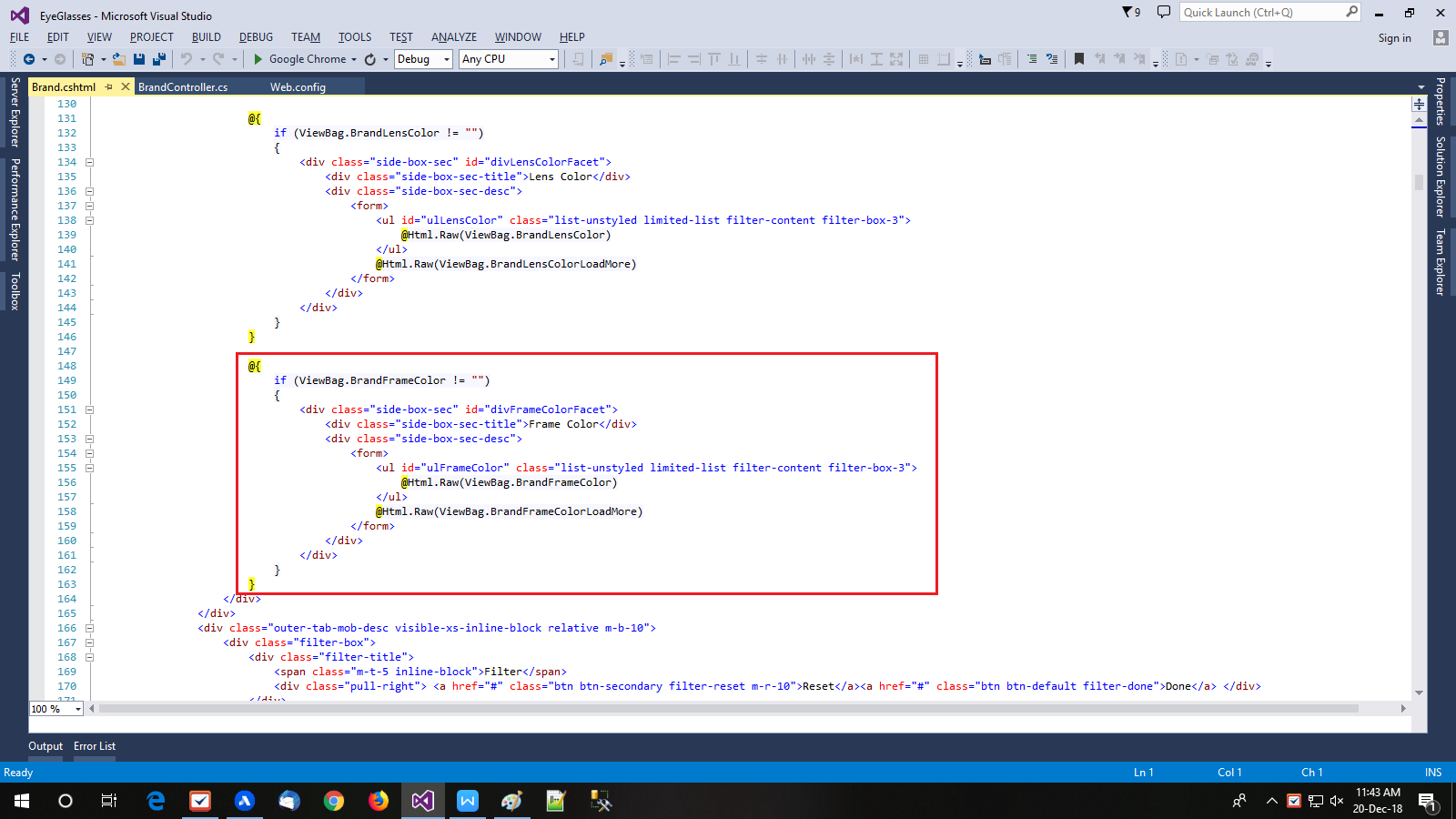
</form>

</div>

</div>

}

}



Now add design for mobile version. Add below code for it. And it’s location illustrate in below figure.

<li id="ddMoFrameColorFacet" class=" dropdown">

<span class="block filter-in-btn">Frame Color <i class="fa fa-plus"></i><i class="fa fa-minus"></i></span>

<div class="inner-tab-desc">

<ul id="ulMoFrameColor" class=" filter-content filter-box-3">

@Html.Raw(ViewBag.BrandFrameColor)

</ul>

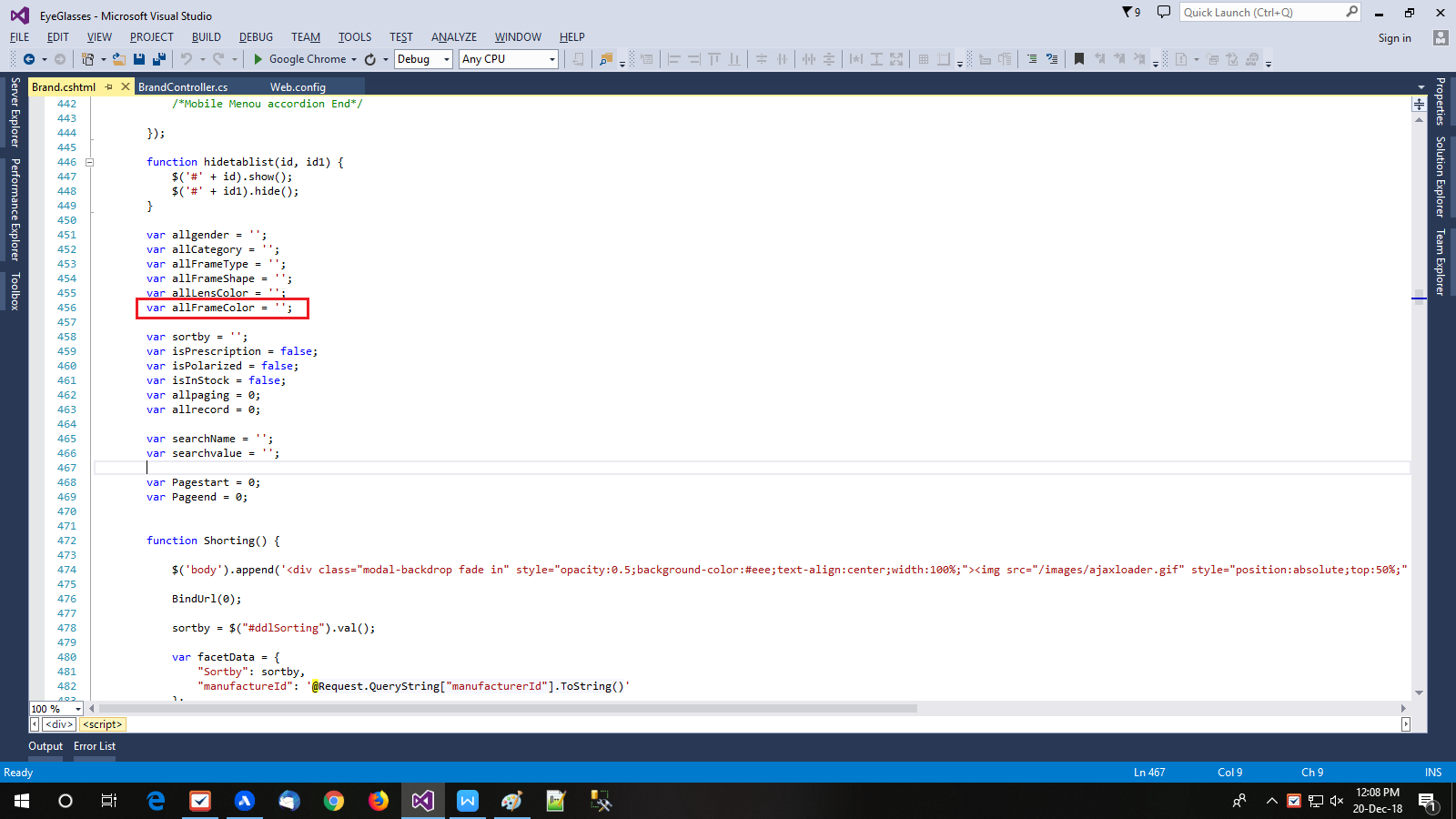
</div>

</li>

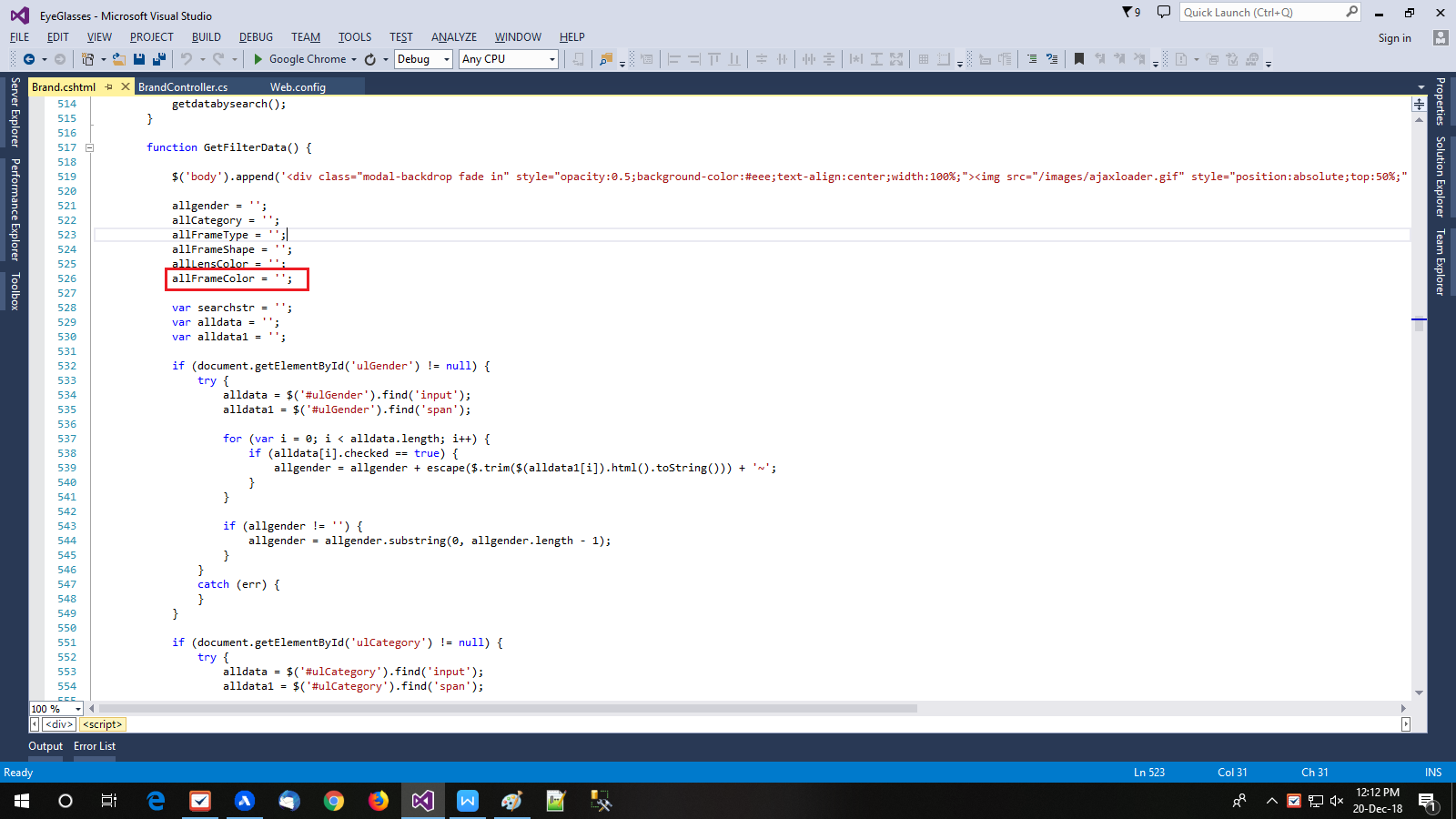
**Step-15 : Add global variable to store filter items from Frame Color**

Add below code at shown in screenshot.

var allFrameColor = '';



**Step-16 : Clear old filter item and store new filter item**



Go tofunction GetFilterData() and add below code as per screenshot to clear old filter items.

allFrameColor = '';

To get new filter items add below code as per shown below screenshot.

if (document.getElementById('ulFrameColor') != null) {

try {

alldata = $('#ulFrameColor').find('input');

alldata1 = $('#ulFrameColor p').find('span');

for (var i = 0; i < alldata.length; i++) {

if (alldata[i].checked == true) {

allFrameColor = allFrameColor + escape($.trim($(alldata1[i]).html().toString())) + '~';

}

}

if (allFrameColor != '') {

allFrameColor = allFrameColor.substring(0, allFrameColor.length - 1);

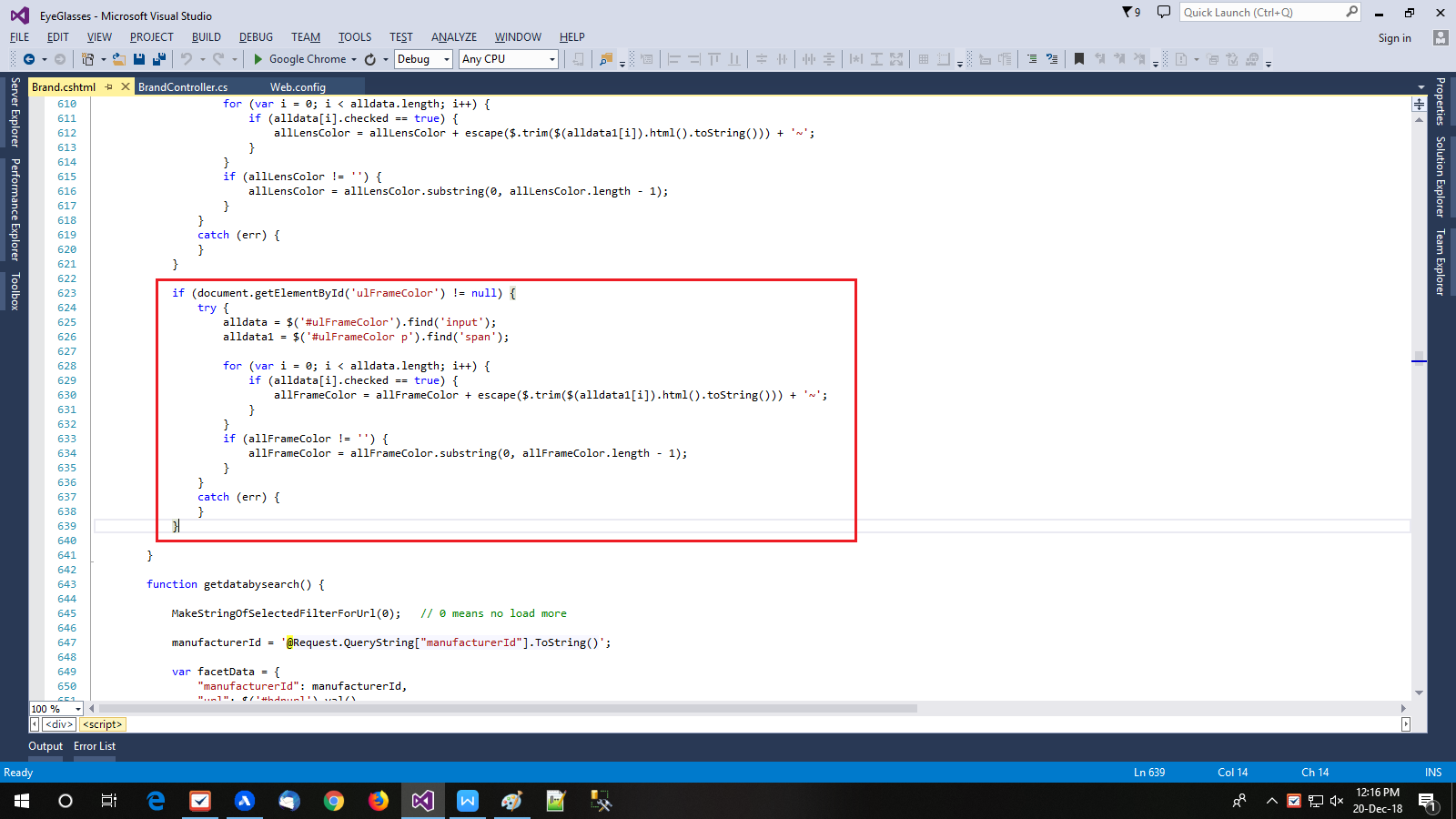
}

}

catch (err) {

}

}

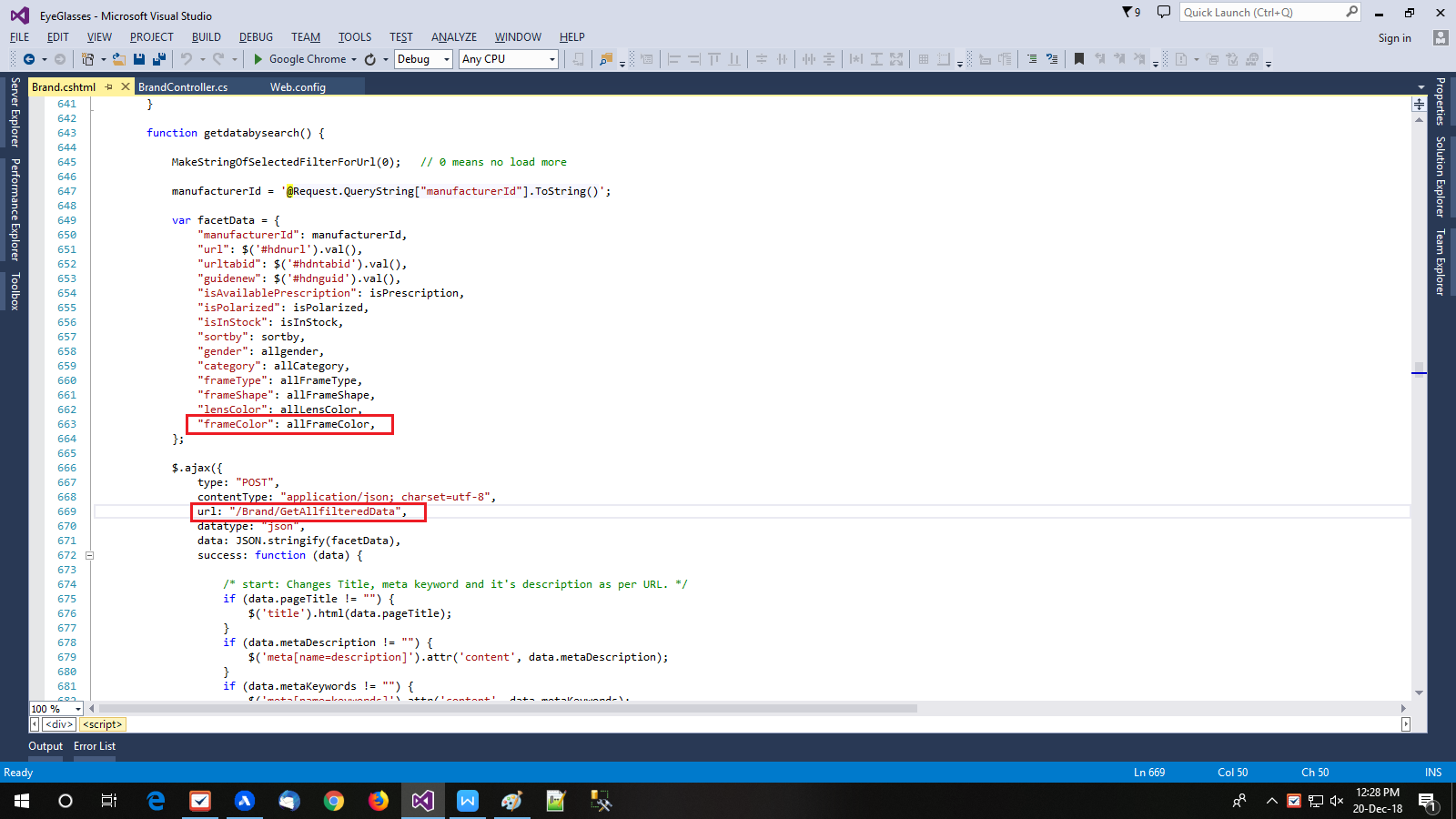


**Step-17 : Add filter items to ajax call**

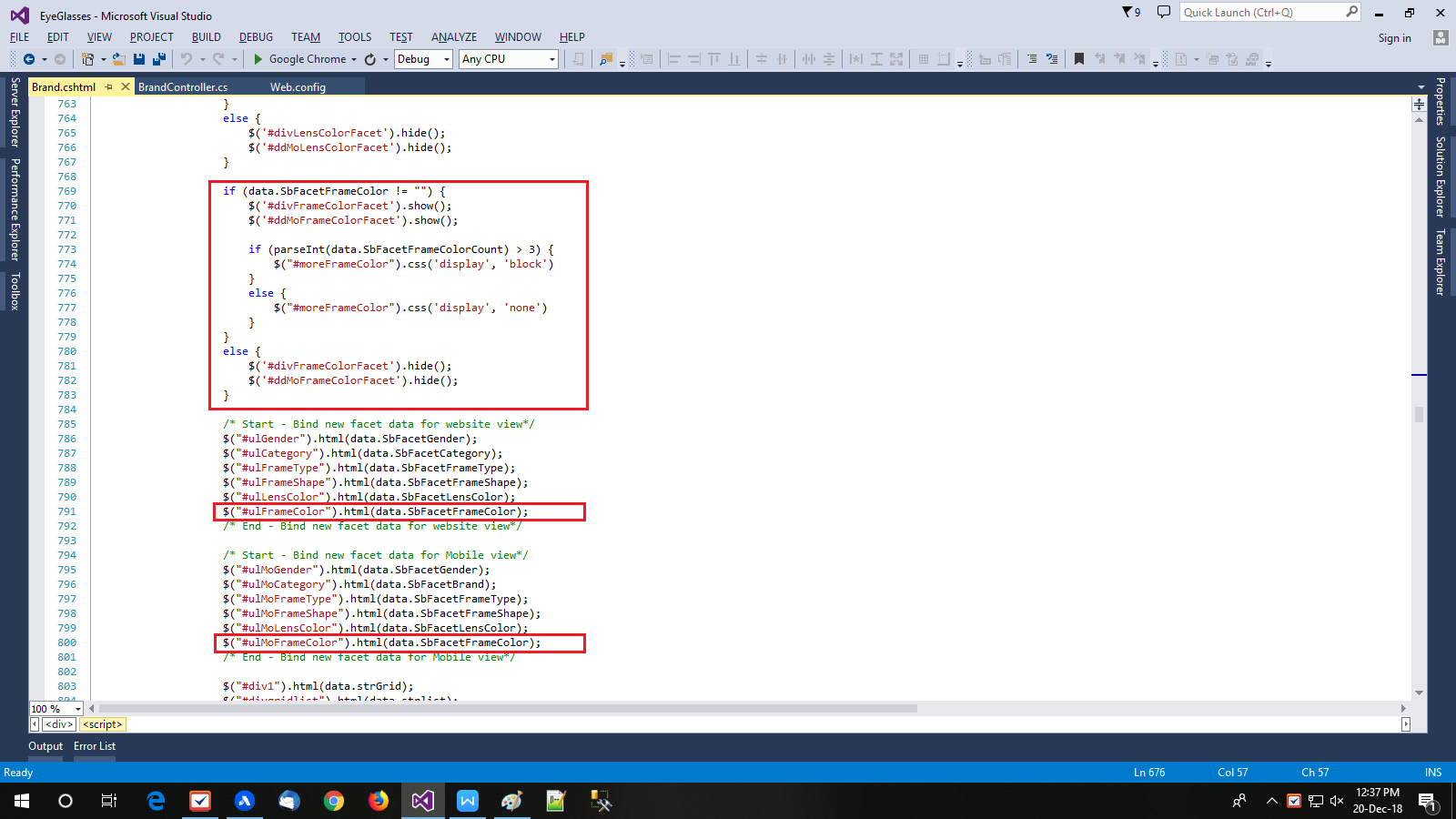
Go to function function getdatabysearch() and add filter items variable to object and pass it to Ajax call as per below.

"frameColor": allFrameColor,

Then it will call method GetAllfilteredData in Brand controller to get new filer HTML design boxes for all filters as per passed new filter items.



Now from it’s success response you can get a new HTML design for filter as per applied filter. So get it from there and set in view using its Id. You can show and hide filter div as per data and set HTML data into its part. Its is illustrate in below screenshot.



**Step-18 : Bind URL as per applied filter**

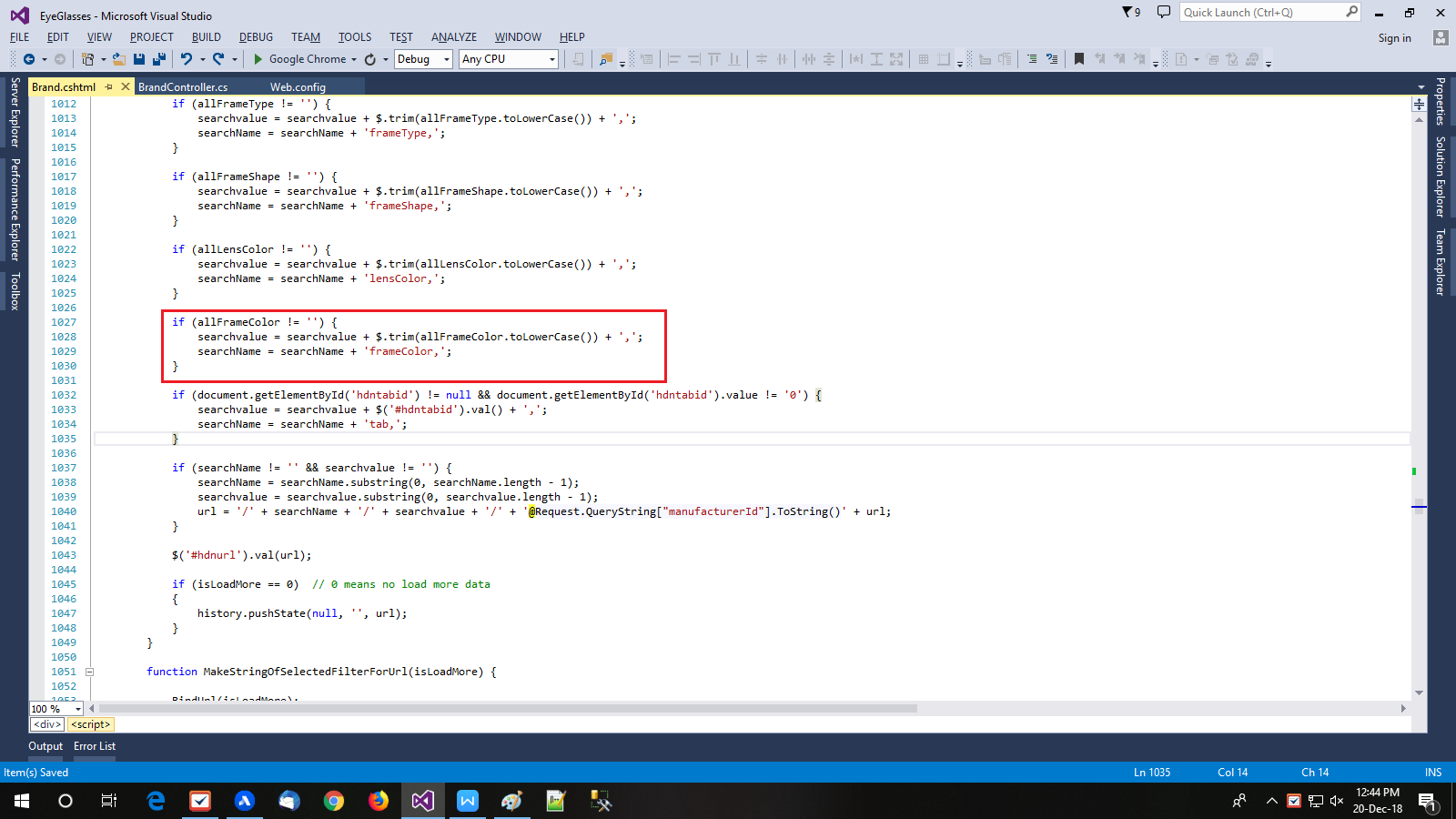
Add below code to add filter data to URL and it’s illustrate in below screenshot.

if (allFrameColor != '') {

searchvalue = searchvalue + $.trim(allFrameColor.toLowerCase()) + ',';

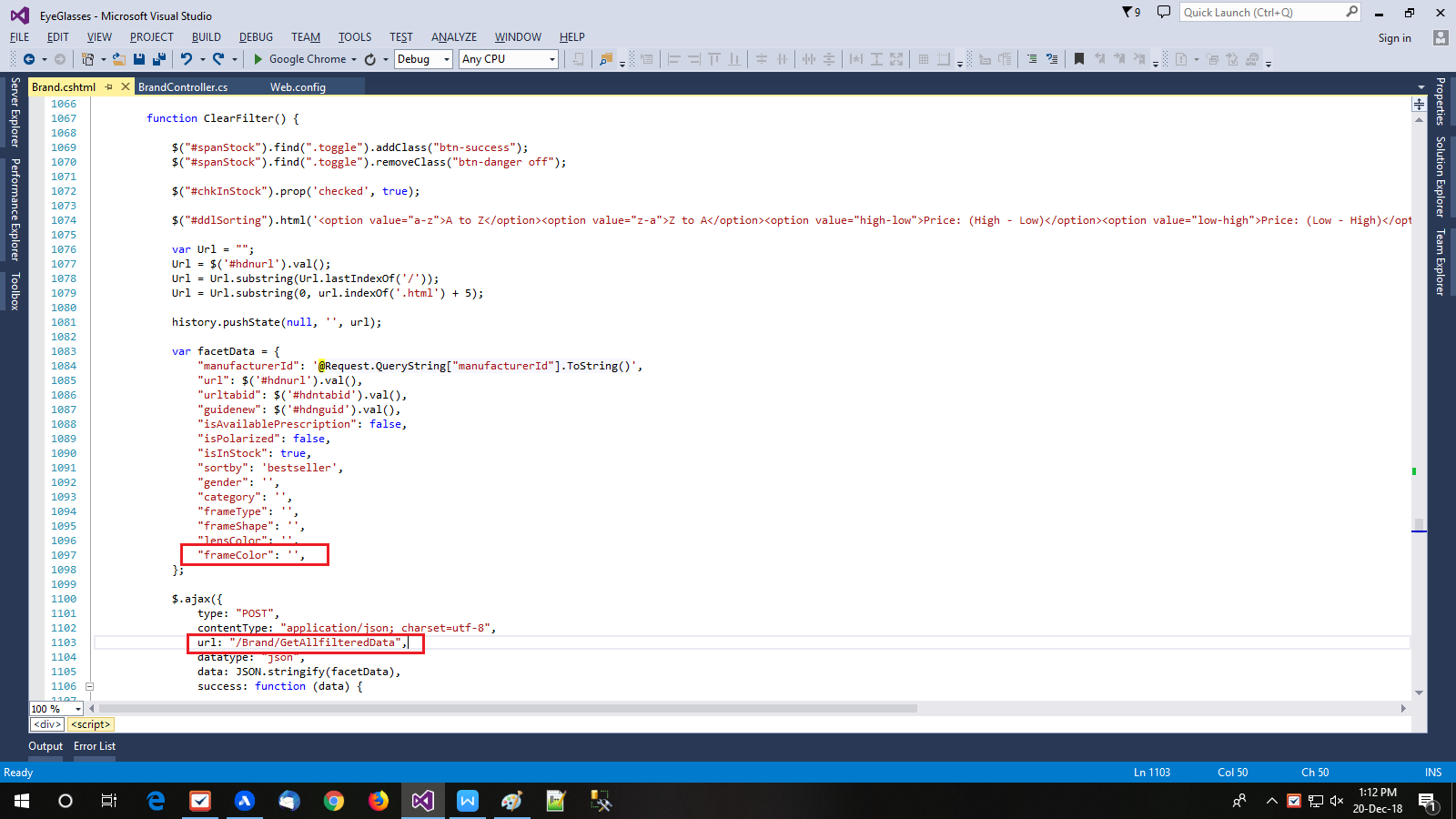
searchName = searchName + 'frameColor,';

}

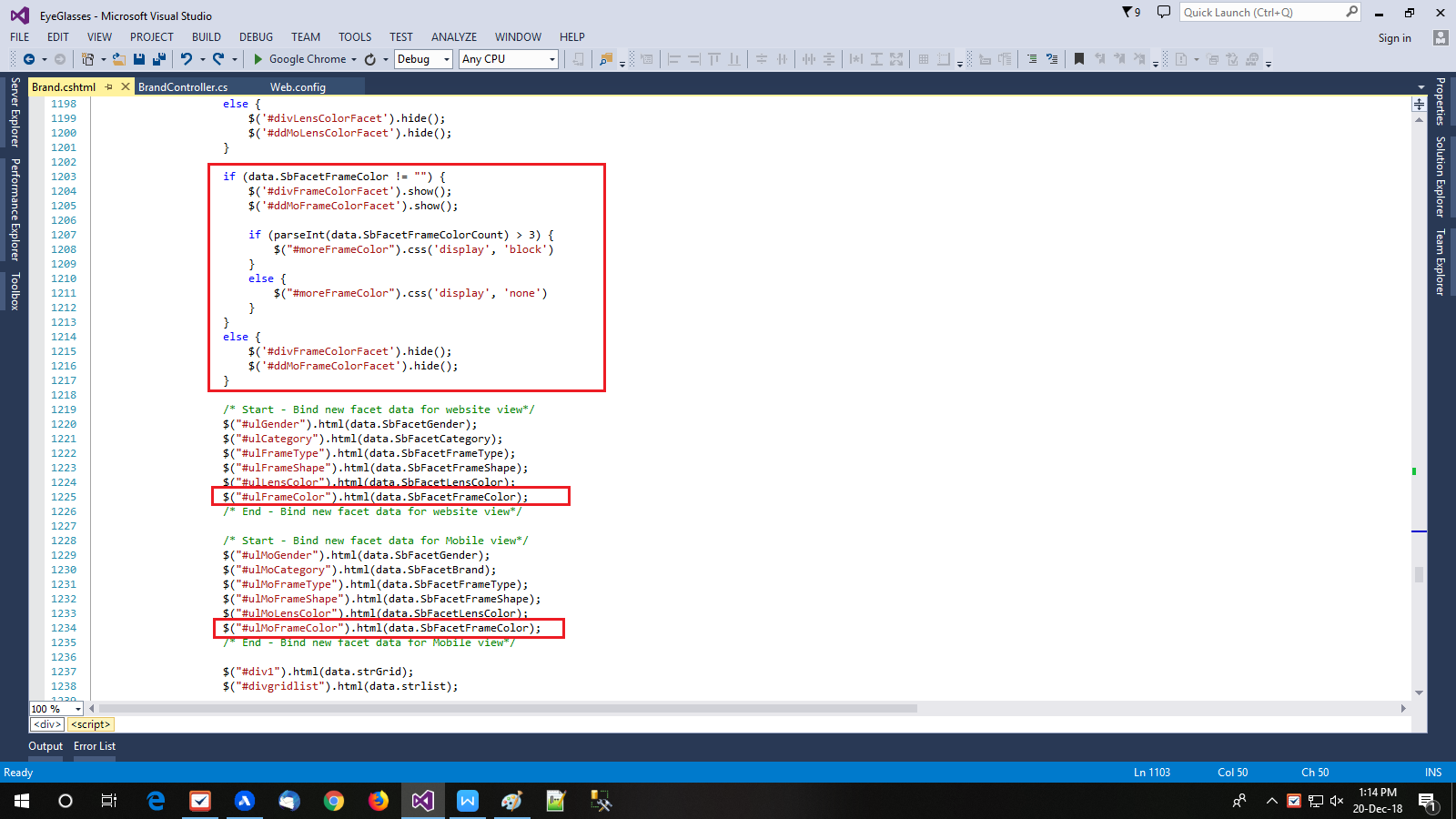


**Step-19 : Clear filters and get data for all**

Clear all filter and get new facet design for all filter from controller by calling method GetAllfilteredData which is illustrate in below screenshot.



Then get new HTML design from success function and set it to its Id which is illustrate in below screenshot.



**Step-20 : Go though methods of BrandController.cs file**

* List of method which are used to get facet navigation HTML design.
* public ActionResult Brand()
* public JsonResult GetAllfilteredData()
* public JsonResult Getpaggingdata()
* private void GetDataFromWebCacheOrDatabase()
* public ArrayList BindFacetData()
* public string FilterProductsbySelf()
* public void FilterProducts()
* public void FilterProductsByFacetData()
* public bool IsProductInCategoryFilter\_Common()
* public void Filteration()
* public JsonResult ShortingProductsAndBinding()
* public void ShortingProducts()
* public void SetMoreProductsForPagging()
* public void ClearMappingData()
* List of method which are used to bind products
* private string ProductBoxlist()
* public JsonResult GetIconImageProduct()
* public JsonResult GetIconBrand()
* private void BindBrandDetails()
* public string Getrating()
* public string GetMenuIconImageBrand()
* public string CalculatePrice()
* Remaining all method are used to manage data in session and cookie as per need.

**Required files and Stored Procedures for Brand and Category Product List page.**

1. **Brand Product List page**

* Pages:
* \EyeGlassFront\EyeGlasses\Controllers\BrandController.cs
* \EyeGlassFront\EyeGlasses\Views\Brand\Brand.cshtml
* Stored Procedures:
* GuiGetBrandFacetedMasterEG
* GuiGetBrandAllSearchdataEG
* GuiGetBrandProductsEG

1. **Category Product List page**

* Pages:
* \EyeGlassFront\EyeGlasses\Controllers\CategoryController.cs
* \EyeGlassFront\EyeGlasses\Views\Category\Category.cshtml
* Stored Procedures:
* GuiGetCategoryFacetedMasterEG
* GuiGetCategoryAllSearchdataEG
* GuiGetCategoryProductsEG