



Education vs. Child Development: How Dyslexia Happens

Having been functionally illiterate for the first 38 years of my life, and having overcome the problem sufficiently to write a book and read it onto tape, here is what I've found out: dyslexia is not the result of brain damage or nerve damage.

[Details »](#)

A Dyslexic Child in the Classroom: What can be done?

A Guide for Teachers and Parents Proficient reading is an essential tool for learning a large part of the subject matter taught at school. With an ever increasing emphasis on education and literacy, more and more children and adults are needing...

[Details »](#)

DYSLEXIA COMBAT SYSTEM

Building a more sensitive community

DECEMBER 19, 2019

Overview

A customer relationship management system to facilitate humans, all ages, facing dyslexia. There will be two modules to the project, the research system and schooling system.



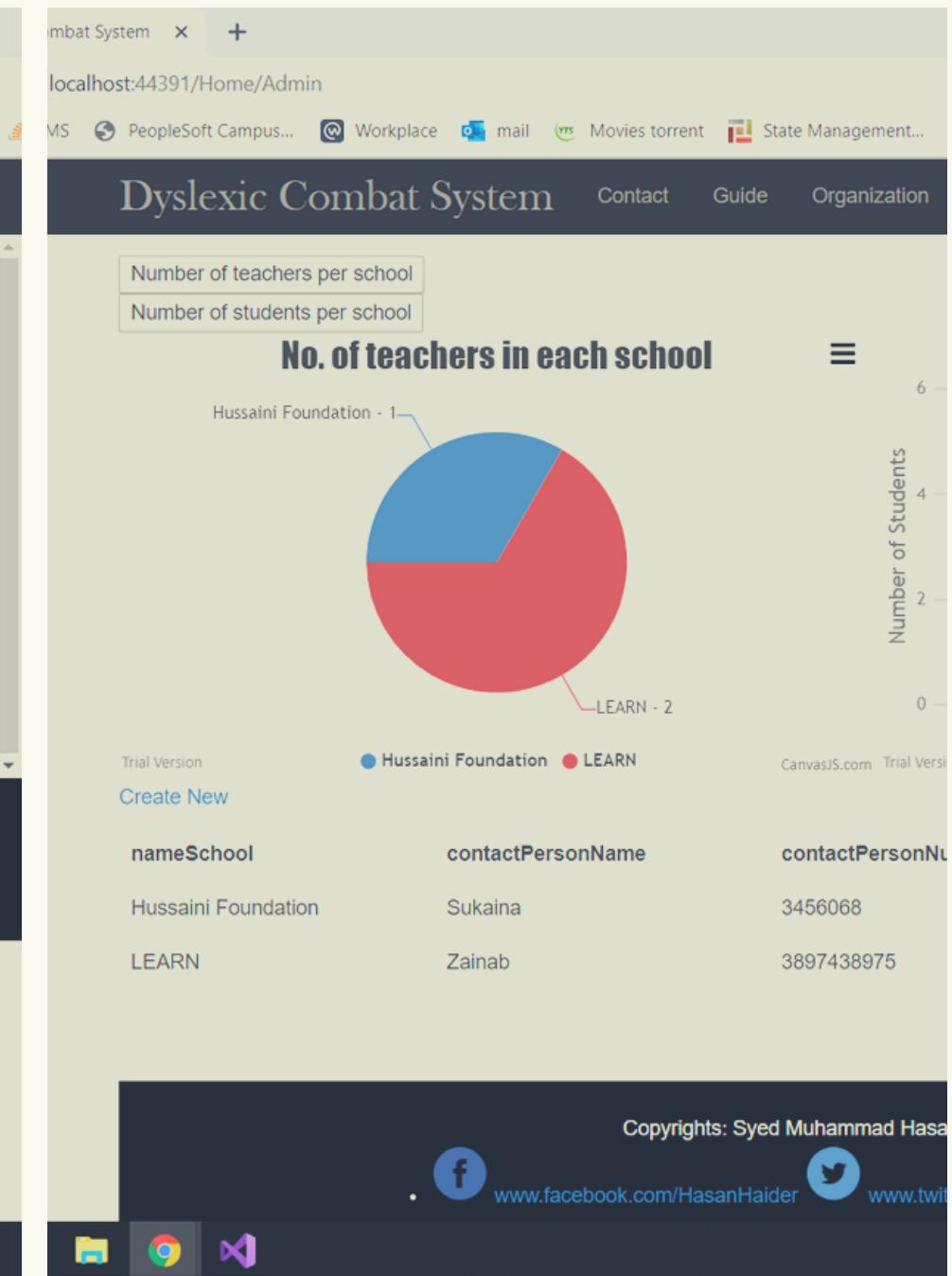
MODULES



USER

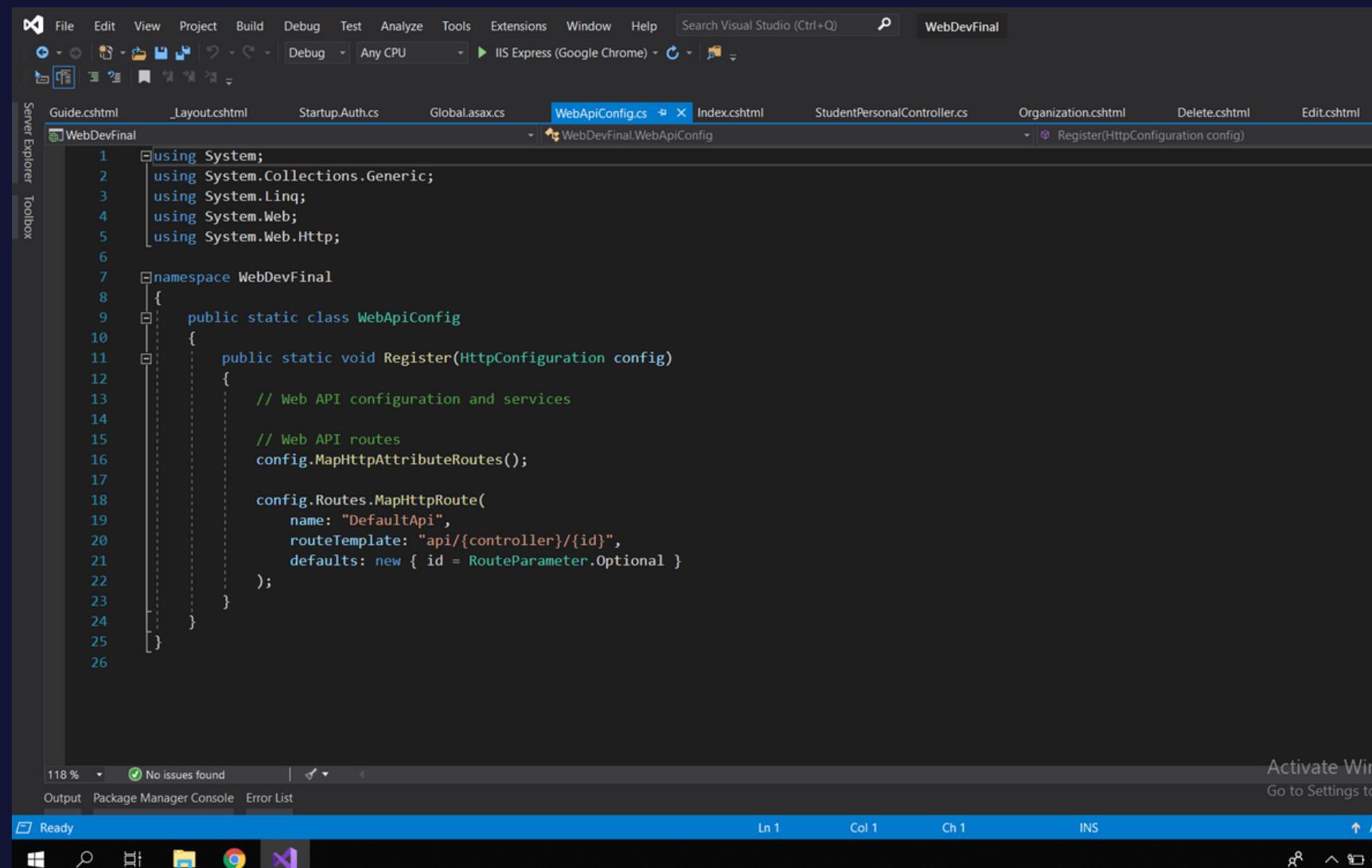


ORGANIZATIONS



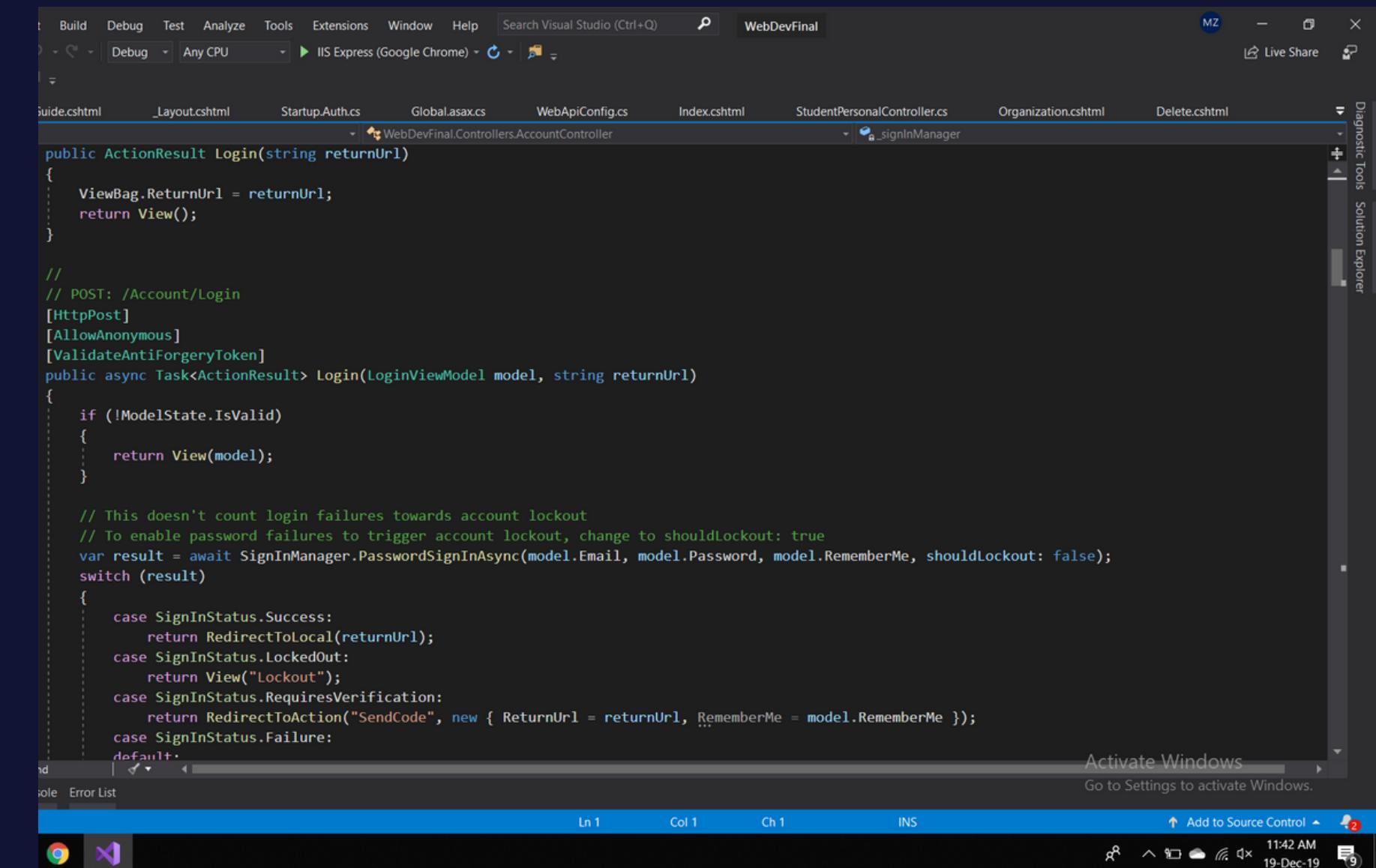
ADMIN

ASP.NET API



The screenshot shows the Visual Studio IDE with the 'WebDevFinal' project open. The 'WebApiConfig.cs' file is the active code editor. The code defines a static class 'WebApiConfig' with a single method 'Register' that configures the Web API routes. The interface includes imports for System, System.Collections.Generic, System.Linq, System.Web, and System.Web.Http. The 'Register' method maps routes for the 'DefaultApi' controller.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Http;
6
7  namespace WebDevFinal
8  {
9      public static class WebApiConfig
10     {
11         public static void Register(HttpConfiguration config)
12         {
13             // Web API configuration and services
14
15             // Web API routes
16             config.MapHttpAttributeRoutes();
17
18             config.Routes.MapHttpRoute(
19                 name: "DefaultApi",
20                 routeTemplate: "api/{controller}/{id}",
21                 defaults: new { id = RouteParameter.Optional }
22             );
23         }
24     }
25 }
```



The screenshot shows the Visual Studio IDE with the 'WebDevFinal.Controllers' folder open. The 'AccountController.cs' file is the active code editor. It contains an 'Login' action method that sets the 'ReturnUrl' view bag and returns a view. It also contains an 'HttpPost' action method 'Login' that handles a POST request to '/Account/Login'. This method checks if the model state is valid, signs in the user using 'SignInManager.PasswordSignInAsync', and then returns a redirect to the specified 'returnUrl'. It handles cases for success, locked out status, and failure.

```
public ActionResult Login(string returnUrl)
{
    ViewBag.ReturnUrl = returnUrl;
    return View();
}

[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }

    // This doesn't count login failures towards account lockout
    // To enable password failures to trigger account lockout, change to shouldLockout: true
    var result = await SignInManager.PasswordSignInAsync(model.Email, model.Password, model.RememberMe, shouldLockout: false);
    switch (result)
    {
        case SignInStatus.Success:
            return RedirectToAction(returnUrl);
        case SignInStatus.LockedOut:
            return View("Lockout");
        case SignInStatus.RequiresVerification:
            return RedirectToAction("SendCode", new { ReturnUrl = returnUrl, RememberMe = model.RememberMe });
        case SignInStatus.Failure:
            default:
    }
}
```

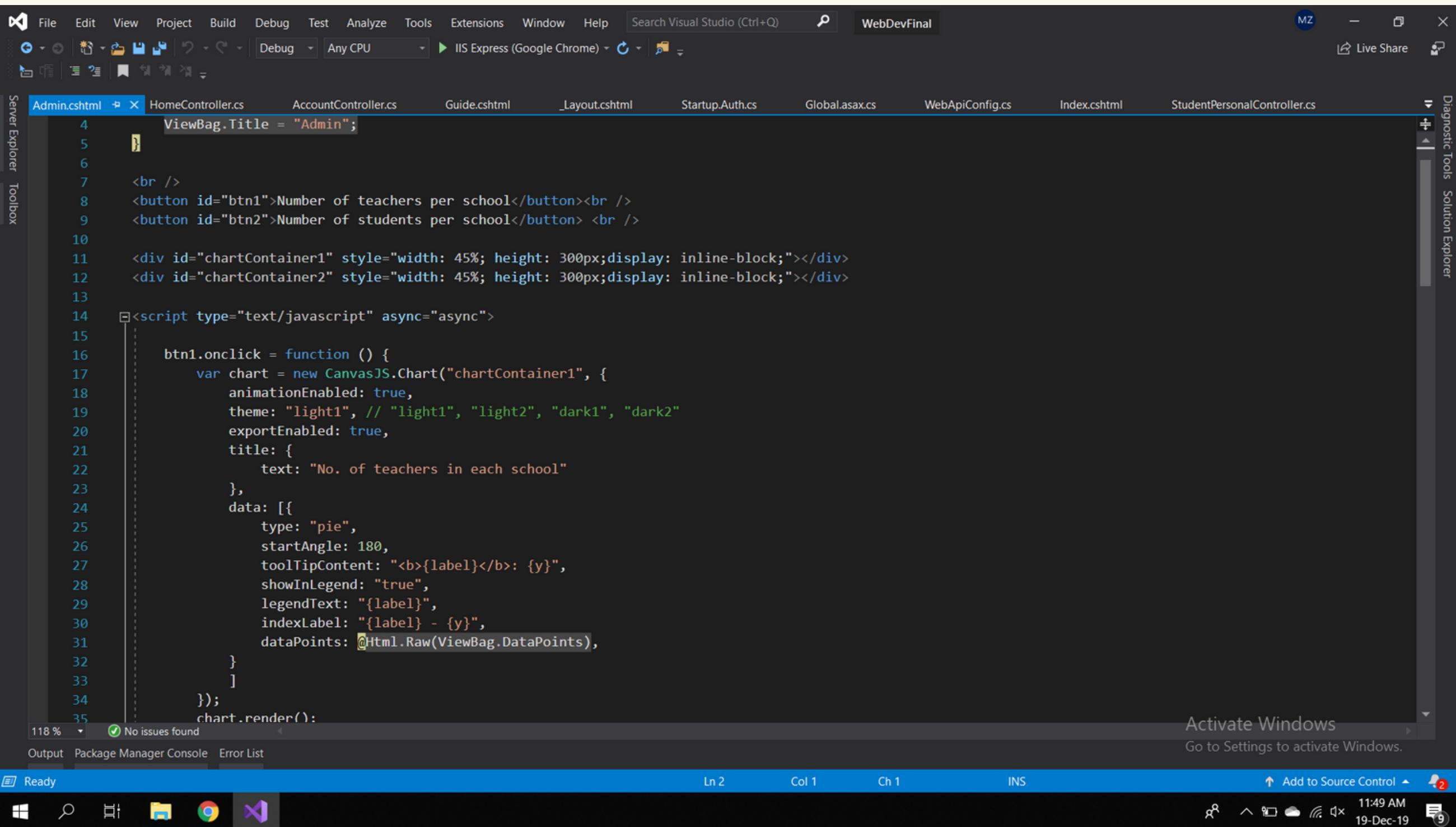
LINQ Queries

The screenshot shows a Microsoft Visual Studio interface with a dark theme. The main window displays a C# code editor for a file named `HomeController.cs` within a project titled `WebDevFinal`. The code implements two actions: `Guide()` and `Organization()`. The `Organization()` action retrieves data from a database using LINQ queries to count grades and add them to a list of `DataPoint`.

```
34 }  
35 [Route("Maham")]  
36 [Route("Home/Guide")]  
37 public ActionResult Guide()  
38 {  
39     ViewBag.Message = "Your CareTaker page.";  
40  
41     return View();  
42 }  
43  
44 [Authorize(Roles = "Organization")]  
45 public ActionResult Organization()  
46 {  
47     var schoolID = (from s in db.Schools  
48                     where s.emailID == User.Identity.Name  
49                     select s.idSchool).First();  
50     var final = (from f in db.StudentPersonals  
51                     where f.School_idSchool == schoolID  
52                     select f);  
53     var gradesCount = (from s in final  
54                     orderby s.grade  
55                     group s by s.grade into grp  
56                     select new { key = grp.Key, cnt = grp.Count() }).ToList();  
57  
58     List<DataPoint> dataPoints = new List<DataPoint>();  
59     //var dataPoints =[];  
60     for (int i = 0; i < gradesCount.Count(); i++)  
61     {  
62         dataPoints.Add(new DataPoint((int)gradesCount[i].key.GetValueOrDefault(), (int)gradesCount[i].cnt));  
63         //Console.log(gradesCount[i]);  
64 }
```

The status bar at the bottom indicates the code is ready, with a 118% zoom level, no issues found, and a timestamp of 11:43 AM on 19-Dec-19.

AJAX



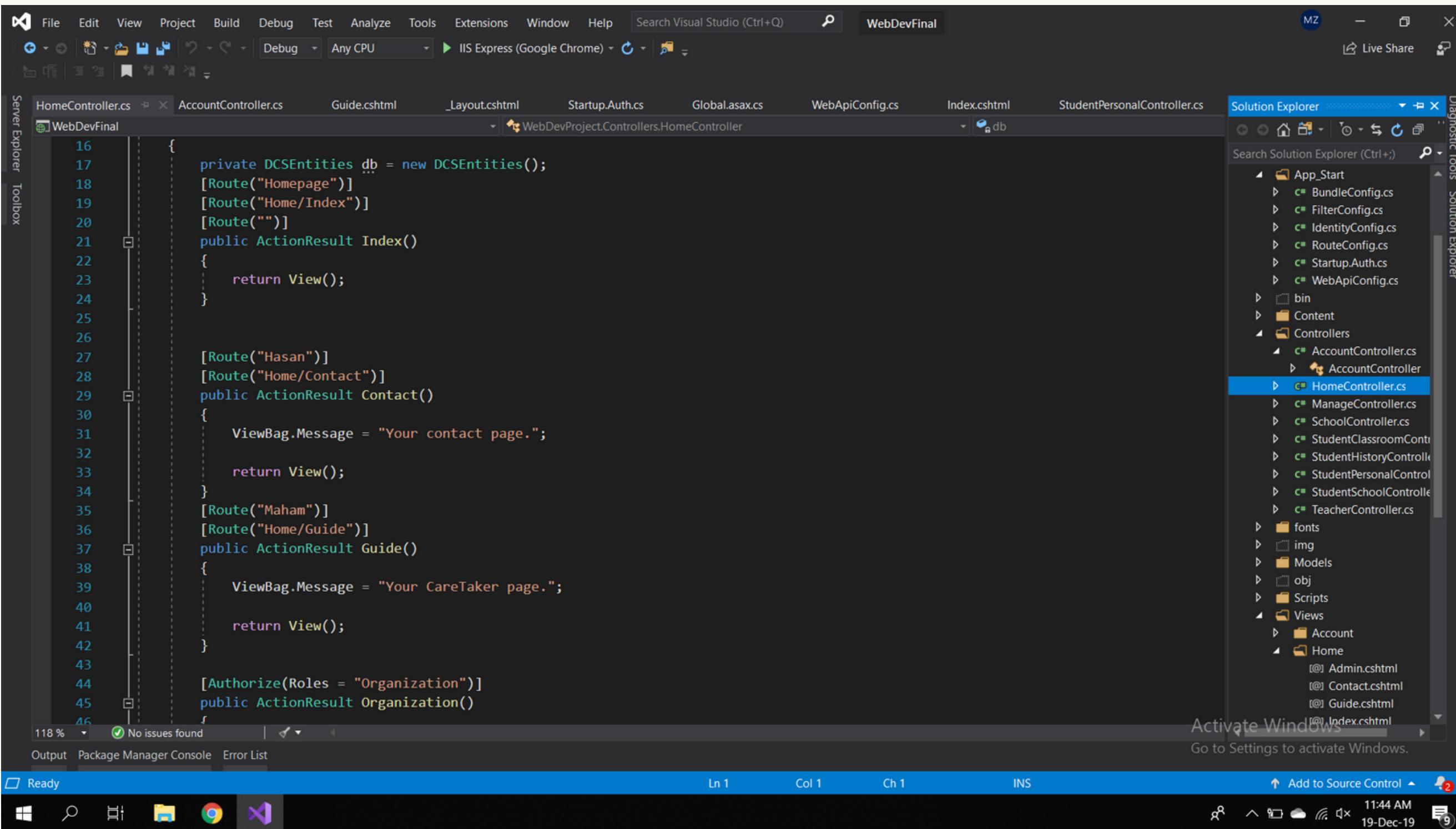
The screenshot shows the Visual Studio IDE interface with the following details:

- Title Bar:** Shows "WebDevFinal" as the project name.
- Toolbars:** Standard File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Status Bar:** Shows "Ready" and system icons like battery level, signal strength, and date/time (11:49 AM, 19-Dec-19).
- Solution Explorer:** Shows files like Admin.cshtml, HomeController.cs, AccountController.cs, Guide.cshtml, _Layout.cshtml, Startup.Auth.cs, Global.asax.cs, WebApiConfig.cs, Index.cshtml, and StudentPersonalController.cs.
- Code Editor:** Displays the Admin.cshtml file content. The code includes HTML for two buttons and two chart containers, followed by a script block that initializes a CanvasJS pie chart on button1 click. The chart configuration includes a title, data type, start angle, tool tip content, show in legend, legend text, index label, and data points from ViewBag.

```
ViewBag.Title = "Admin";  
  
<br />  
<button id="btn1">Number of teachers per school</button><br />  
<button id="btn2">Number of students per school</button> <br />  
  
<div id="chartContainer1" style="width: 45%; height: 300px; display: inline-block;"></div>  
<div id="chartContainer2" style="width: 45%; height: 300px; display: inline-block;"></div>  
  
<script type="text/javascript" async="async">  
    btn1.onclick = function () {  
        var chart = new CanvasJS.Chart("chartContainer1", {  
            animationEnabled: true,  
            theme: "light1", // "light1", "light2", "dark1", "dark2"  
            exportEnabled: true,  
            title: {  
                text: "No. of teachers in each school"  
            },  
            data: [{  
                type: "pie",  
                startAngle: 180,  
                toolTipContent: "<b>{label}</b>: {y}",  
                showInLegend: "true",  
                legendText: "{label}",  
                indexLabel: "{label} - {y}",  
                dataPoints: @Html.Raw(ViewBag.DataPoints),  
            }]  
        });  
        chart.render();  
    };  
</script>
```

- Output:** Shows "No issues found".
- Activate Windows:** A watermark at the bottom right.

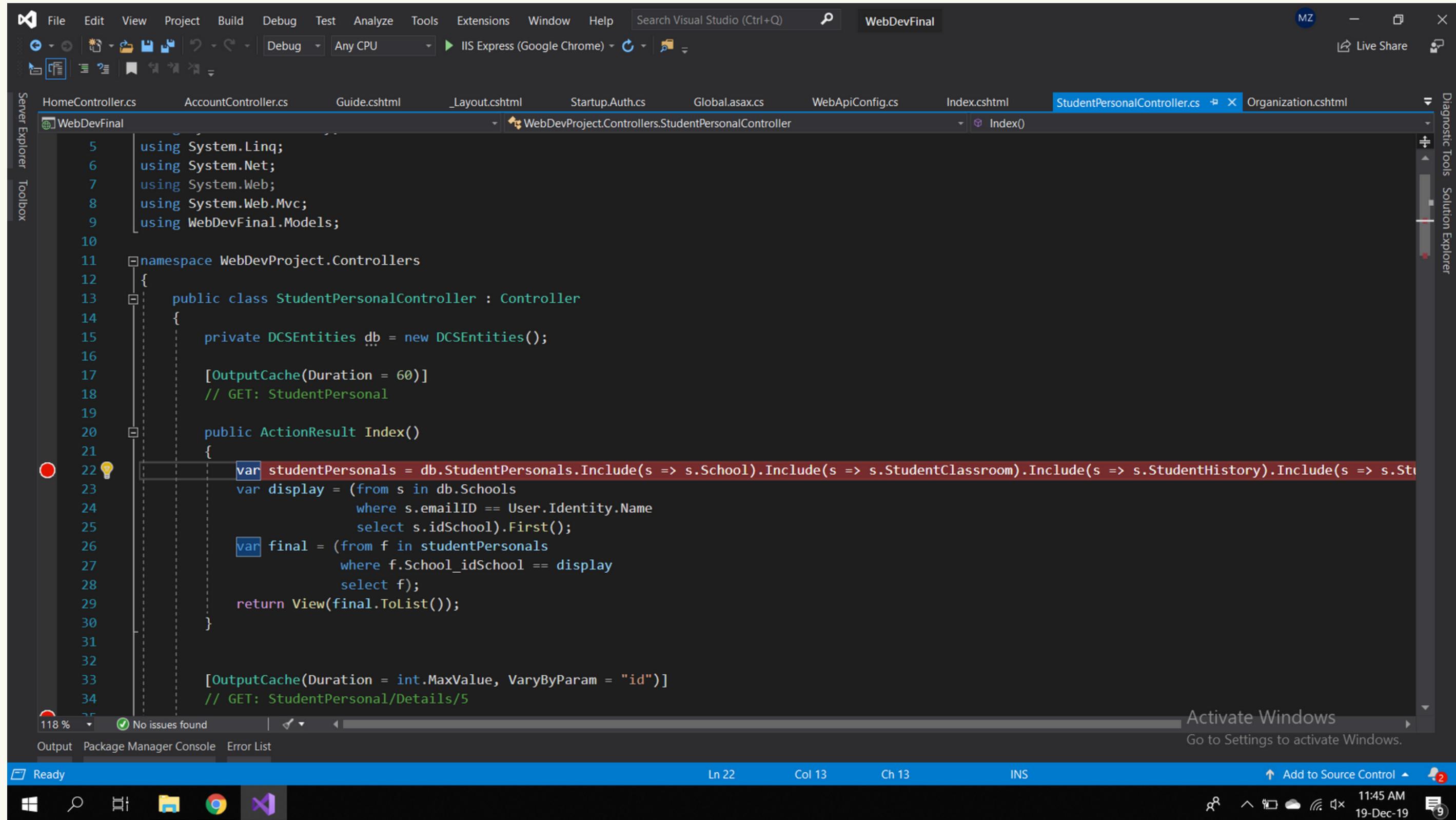
Custom Routing



The screenshot shows the Visual Studio IDE interface with the following details:

- MenuBar:** File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search Visual Studio (Ctrl+Q).
- Toolbars:** Standard toolbar with icons for New, Open, Save, Print, etc.
- Code Editor:** Displays the `HomeController.cs` file content. The code defines several actions with custom routes:
 - `[Route("Homepage")]`
 - `[Route("Home/Index")]`
 - `[Route("")]` (Index action)
 - `[Route("Hasan")]`
 - `[Route("Home/Contact")]`
 - `[Route("Maham")]`
 - `[Route("Home/Guide")]`
 - `[Authorize(Roles = "Organization")]` (Organization action)
- Solution Explorer:** Shows the project structure:
 - App_Start
 - bin
 - Content
 - Controllers
 - AccountController.cs
 - HomeController.cs (selected)
 - ManageController.cs
 - SchoolController.cs
 - StudentClassroomController
 - StudentHistoryController
 - StudentPersonalController
 - StudentSchoolController
 - TeacherController.cs
 - fonts
 - img
 - Models
 - obj
 - Scripts
 - Views
 - Account
 - Home
 - Admin.cshtml
 - Contact.cshtml
 - Guide.cshtml
 - Index.cshtml
- Status Bar:** Shows the status bar with various icons and the message "Activate Windows".
- Taskbar:** Shows the taskbar with icons for File Explorer, Task View, Start, Search, and Visual Studio.

Caching

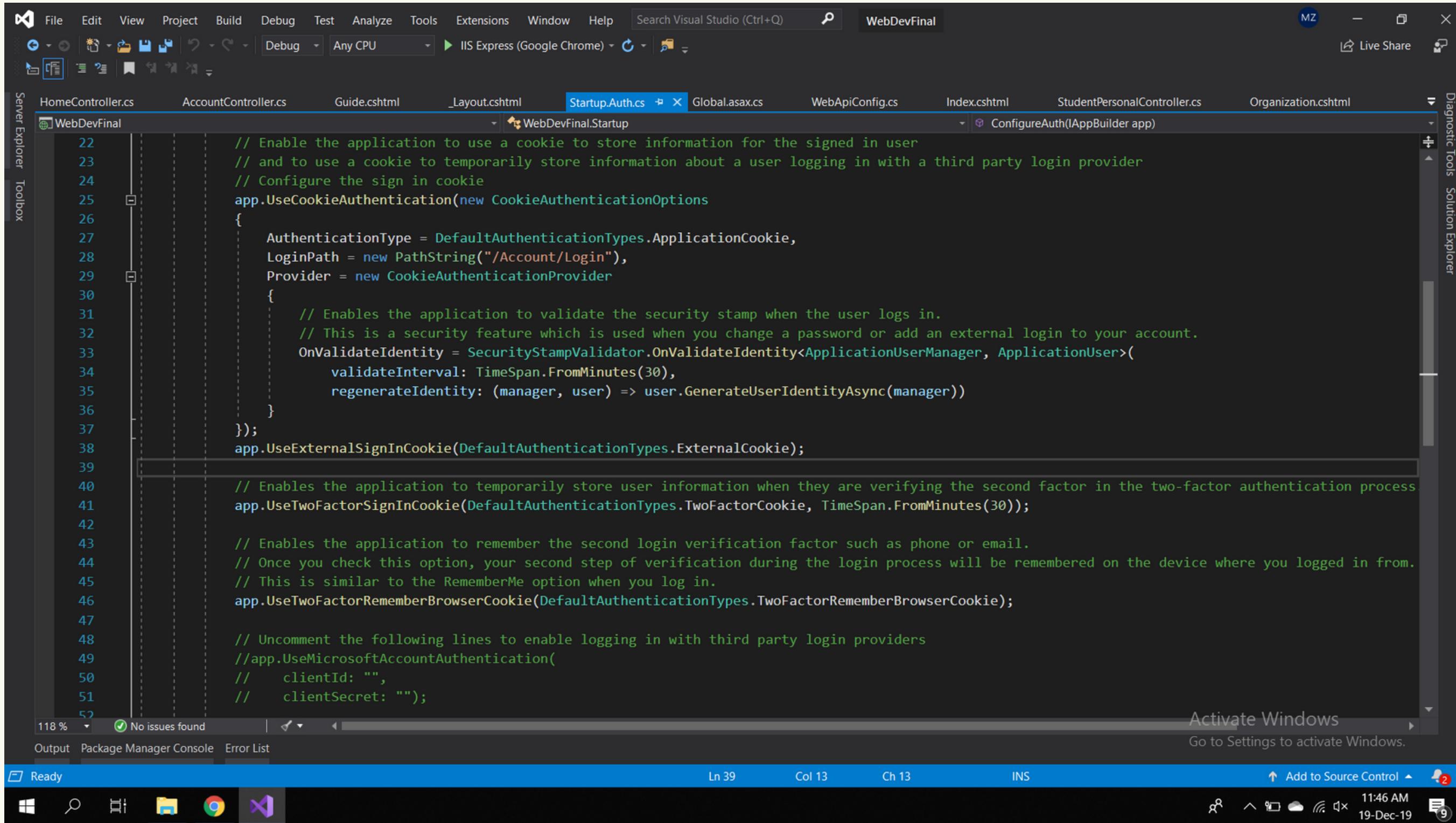


The screenshot shows a Visual Studio IDE window with the following details:

- Title Bar:** Shows "WebDevFinal" as the project name.
- Toolbars:** Standard Visual Studio toolbars for File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and a Search bar.
- Solution Explorer:** Shows files like HomeController.cs, AccountController.cs, Guide.cshtml, _Layout.cshtml, Startup.Auth.cs, Global.asax.cs, WebApiConfig.cs, Index.cshtml, StudentPersonalController.cs (selected), and Organization.cshtml.
- Code Editor:** Displays the "StudentPersonalController.cs" file. The code implements a controller for student personal data, using Entity Framework's Include method to eager load multiple related entities. A red circle and a yellow lightbulb icon are positioned near the start of the code block.
- Status Bar:** Shows "Activate Windows" and "Go to Settings to activate Windows." It also displays zoom level (118%), issue status ("No issues found"), and navigation information (Ln 22, Col 13, Ch 13, INS).
- Taskbar:** Shows icons for File, Search, Task View, File Explorer, Google Chrome, and Visual Studio.
- System Tray:** Shows system icons for battery, signal, volume, and date/time (11:45 AM, 19-Dec-19).

```
5  using System.Linq;
6  using System.Net;
7  using System.Web;
8  using System.Web.Mvc;
9  using WebDevFinal.Models;
10
11 namespace WebDevProject.Controllers
12 {
13     public class StudentPersonalController : Controller
14     {
15         private DCSEntities db = new DCSEntities();
16
17         [OutputCache(Duration = 60)]
18         // GET: StudentPersonal
19
20         public ActionResult Index()
21         {
22             var studentPersonals = db.StudentPersonals.Include(s => s.School).Include(s => s.StudentClassroom).Include(s => s.StudentHistory).Include(s => s.St
23             var display = (from s in db.Schools
24                           where s.emailID == User.Identity.Name
25                           select s.idSchool).First();
26             var final = (from f in studentPersonals
27                           where f.School_idSchool == display
28                           select f);
29             return View(final.ToList());
30         }
31
32         [OutputCache(Duration = int.MaxValue, VaryByParam = "id")]
33         // GET: StudentPersonal/Details/5
34     }
```

Cookies



The screenshot shows the Visual Studio IDE interface with the following details:

- Title Bar:** Shows "WebDevFinal" as the project name.
- Toolbars:** Standard Visual Studio toolbars for File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** "Search Visual Studio (Ctrl+Q)"
- Status Bar:** Shows "Ready", "Ln 39", "Col 13", "Ch 13", "INS", "Add to Source Control", and a date/time stamp "11:46 AM 19-Dec-19".
- Code Editor:** Displays the "Startup.Auth.cs" file with C# code for cookie authentication. The code includes comments explaining the use of ApplicationCookie, TwoFactorCookie, and TwoFactorRememberBrowserCookie.
- Solution Explorer:** Shows files like HomeController.cs, AccountController.cs, Guide.cshtml, _Layout.cshtml, Global.asax.cs, WebApiConfig.cs, Index.cshtml, StudentPersonalController.cs, and Organization.cshtml.
- Server Explorer:** Shows the "WebDevFinal" connection status.
- Output, Package Manager Console, Error List:** Standard Visual Studio tabs at the bottom.

```
22     // Enable the application to use a cookie to store information for the signed in user
23     // and to use a cookie to temporarily store information about a user logging in with a third party login provider
24     // Configure the sign in cookie
25     app.UseCookieAuthentication(new CookieAuthenticationOptions
26     {
27         AuthenticationType = DefaultAuthenticationTypes.ApplicationCookie,
28         LoginPath = new PathString("/Account/Login"),
29         Provider = new CookieAuthenticationProvider
30         {
31             // Enables the application to validate the security stamp when the user logs in.
32             // This is a security feature which is used when you change a password or add an external login to your account.
33             OnValidateIdentity = SecurityStampValidator.OnValidateIdentity<ApplicationUserManager, ApplicationUser>(
34                 validateInterval: TimeSpan.FromMinutes(30),
35                 regenerateIdentity: (manager, user) => user.GenerateUserIdentityAsync(manager))
36         }
37     });
38     app.UseExternalSignInCookie(DefaultAuthenticationTypes.ExternalCookie);
39
40     // Enables the application to temporarily store user information when they are verifying the second factor in the two-factor authentication process
41     app.UseTwoFactorSignInCookie(DefaultAuthenticationTypes.TwoFactorCookie, TimeSpan.FromMinutes(30));
42
43     // Enables the application to remember the second login verification factor such as phone or email.
44     // Once you check this option, your second step of verification during the login process will be remembered on the device where you logged in from.
45     // This is similar to the RememberMe option when you log in.
46     app.UseTwoFactorRememberBrowserCookie(DefaultAuthenticationTypes.TwoFactorRememberBrowserCookie);
47
48     // Uncomment the following lines to enable logging in with third party login providers
49     //app.UseMicrosoftAccountAuthentication(
50     //    clientId: "",
51     //    clientSecret: "");
```

CSS

A screenshot of the Microsoft Visual Studio IDE interface. The main window shows the code editor with a file named "Index.cshtml". The code is a partial view for a web application, featuring a carousel component. The editor has syntax highlighting for HTML and CSS. On the left, the Solution Explorer shows files like HomeController.cs, AccountController.cs, Guide.cshtml, _Layout.cshtml, Startup.Auth.cs, Global.asax.cs, WebApiConfig.cs, StudentPersonalController.cs, and Organization.cshtml. The status bar at the bottom indicates "Ready" and shows the current time as 11:47 AM on 19-Dec-19.

```
13 <p style="font: 15px">As a result they have not developed diagnostic tests that look for the talents and strengths that are typically also part of dyslexia</p>
14 
15 
16 
17 </div>
18 <div col-md-12 class="row" style="color:black;padding-top:10px;text-align:center; background-image:url('/img/bg1.jpg')">
19 <p><a href="/Views/Account/Register.cshtml" class="btn btn-primary btn-lg">Signup &raquo;</a></p>
20 </div>
21 
22 <h1></h1>
23 <div class="container row" style="align-content:center; padding-top:5%;padding-bottom:5%">
24 <div col-md-12>
25 <div id="myCarousel" class="carousel slide" data-ride="carousel">
26 <!-- Indicators -->
27 <ol class="carousel-indicators">
28 <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
29 <li data-target="#myCarousel" data-slide-to="1"></li>
30 <li data-target="#myCarousel" data-slide-to="2"></li>
31 </ol>
32 
33 <!-- Wrapper for slides -->
34 <div class="carousel-inner">
35 <div class="item active">
36 
37 </div>
38 
39 <div class="item">
40 
41 </div>
42 
43 <div class="item">
44 
```

No issues found

Output Package Manager Console Error List

Ln 10 Col 46 Ch 46 INS

11:47 AM 19-Dec-19 9

Screenshot saved
The screenshot was added to your
Activate Windows
OneDrive.
Go to Settings to activate Windows.
OneDrive

Authorization

The screenshot shows the Visual Studio IDE interface with the following details:

- Title Bar:** Shows "WebDevFinal" as the project name.
- Toolbars:** Standard Visual Studio toolbars for File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** "Search Visual Studio (Ctrl+Q)"
- Status Bar:** Shows "Ready", "Ln 49", "Col 55", "Ch 55", "INS", "Add to Source Control", "11:46 AM", "19-Dec-19", and a notification icon with "2".
- Solution Explorer:** Shows files like HomeController.cs, AccountController.cs, Guide.cshtml, _Layout.cshtml, Startup.Auth.cs, Global.asax.cs, WebApiConfig.cs, Index.cshtml, StudentPersonalController.cs, and Organization.cshtml.
- Code Editor:** Displays the following C# code from HomeController.cs:

```
37     public ActionResult Guide()
38     {
39         ViewBag.Message = "Your CareTaker page.";
40
41         return View();
42     }
43
44     [Authorize(Roles = "Organization")]
45     public ActionResult Organization()
46     {
47         var schoolID = (from s in db.Schools
48                         where s.emailID == User.Identity.Name
49                         select s.idSchool).First();
50
51         var final = (from f in db.StudentPersonals
52                         where f.School_idSchool == schoolID
53                         select f);
54
55         var gradesCount = (from s in final
56                         orderby s.grade
57                         group s by s.grade into grp
58                         select new { key = grp.Key, cnt = grp.Count() }).ToList();
59
60         List<DataPoint> dataPoints = new List<DataPoint>();
61         //var dataPoints =[];
62         for (int i = 0; i < gradesCount.Count(); i++)
63         {
64             dataPoints.Add(new DataPoint((int)gradesCount[i].key.GetValueOrDefault(), (int)gradesCount[i].cnt));
65             //Console.log(gradesCount[i]);
66         }
67
68         ViewBag.DataPoints = JsonConvert.SerializeObject(dataPoints);
69         return View();
70     }
```
- Output Bar:** Shows "118 %", "No issues found", "Activate Windows", and "Go to Settings to activate Windows."
- Taskbar:** Icons for Windows, Search, Task View, File Explorer, Google Chrome, and Visual Studio.