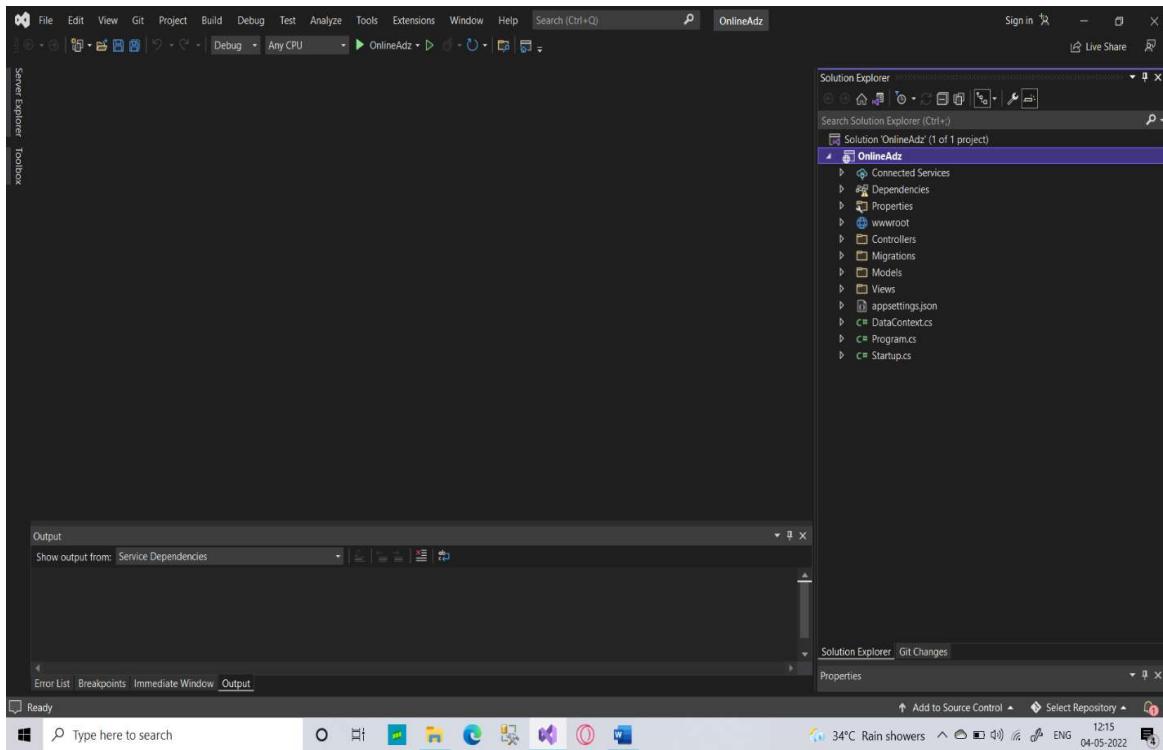


# **Online Ads**

## Basic structure of the Project:



## Controllers: AccountController

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `AccountController.cs` file under the `OnlineAdz` project. The code implements the `AccountController` class, which injects `UserManager<IdentityUser>` and `SignInManager<IdentityUser>` via constructor injection. It contains a `Register` action method that returns a view and an asynchronous `HttpPost` handler for the same action. The `HttpPost` handler checks if the model state is valid, creates a new `IdentityUser`, and performs registration. The `Output` window at the bottom shows "No issues found". The `Solution Explorer` on the right lists all files in the `OnlineAdz` project, including `AccountController.cs`, `CategoryAdController.cs`, `HomeController.cs`, `PlatformController.cs`, `UserAdController.cs`, `UserInterestController.cs`, `Migrations`, `Models`, `Views`, `appsettings.json`, `DataContext.cs`, `Programs`, and `Startup.cs`.

```
using OnlineAdz.Models;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using System.Linq;
using System.Threading.Tasks;

namespace OnlineAdz.Controllers
{
    [Authorize]
    public class AccountController : Controller
    {
        private readonly UserManager<IdentityUser> _userManager;
        private readonly SignInManager<IdentityUser> _signInManager;
        public AccountController(UserManager<IdentityUser> userManager, SignInManager<IdentityUser> signInManager)
        {
            _userManager = userManager;
            _signInManager = signInManager;
        }

        [HttpGet]
        public IActionResult Register()
        {
            return View();
        }

        [HttpPost]
        public async Task<ActionResult> Register(RegisterViewModel model)
        {
            if (ModelState.IsValid)
            {
                var user = new IdentityUser
                {

```

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `AccountController.cs` file under the `OnlineAdz` project. The code implements a `Logout` method that returns a redirect to the `Login` action of the `Account` controller. The `ModelState.AddModelError` line has a tooltip indicating it is used to add validation errors to the model state.

```
        public IActionResult Logout()
        {
            return RedirectToAction("Login", "Account");
        }
    }
```

The Solution Explorer on the right shows the project structure for `OnlineAdz`, which includes files like `CategoryAdminController.cs`, `HomeController.cs`, `PlatformController.cs`, `UserAdController.cs`, `UserInterestController.cs`, `Startup.cs`, and `Program.cs`. The `Controllers` folder contains `AccountController.cs` and `CategoryAdminController.cs`.

Controllers:CategoryAdController:

```
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) OnlineAdz Sign In Live Share

CategoryAdController.cs x
OnlineAdz OnlineAdz.Controllers.CategoryAdController context

using Microsoft.AspNetCore.Mvc;
using OnlineAdz.Models;

namespace OnlineAdz.Controllers
{
    public class CategoryAdController : Controller
    {
        private readonly ApplicationDbContext context = new ApplicationDbContext();
        public IActionResult Index()
        {
            return View(context.CategoryAds.ToList());
        }
        public IActionResult Create()
        {
            return View();
        }
        [HttpPost]
        public IActionResult Create(CategoryAd cad)
        {
            if (cad != null)
            {
                context.CategoryAds.Add(cad);
                context.SaveChanges();
                return RedirectToAction("Index");
            }
            return View();
        }
    }
}
```

Output  
Show output from: Service Dependencies

Error List Breakpoints Immediate Window Output

Ready

Type here to search

Add to Source Control Select Repository

AWL -5.00% 12:31 ENG 04/05/2022

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `CategoryAdController.cs` file under the `OnlineAdz` project. The code implements the `IController` interface with methods for editing and deleting category ads. The Solution Explorer on the right shows the project structure, including controllers like `AccountController.cs`, `CategoryAdController.cs`, and `HomeController.cs`. The status bar at the bottom indicates the file is saved.

```
public class CategoryAdController : Controller
{
    private readonly ApplicationDbContext context;

    public CategoryAdController()
    {
        context = new ApplicationDbContext();
    }

    [HttpGet]
    public ActionResult Index()
    {
        var categories = context.CategoryAds.ToList();
        return View(categories);
    }

    [HttpGet]
    public ActionResult Edit(string id)
    {
        var category = context.CategoryAds.Find(id);
        if (category == null)
        {
            return HttpNotFound();
        }
        return View(category);
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit(CategoryAd category)
    {
        if (category == null)
        {
            return HttpNotFound();
        }
        context.CategoryAds.Update(category);
        context.SaveChanges();
        return RedirectToAction("Index");
    }

    [HttpGet]
    public ActionResult Delete(string id)
    {
        var category = context.CategoryAds.Find(id);
        if (category == null)
        {
            return HttpNotFound();
        }
        context.CategoryAds.Remove(category);
        context.SaveChanges();
        return RedirectToAction("Index");
    }

    [HttpGet]
    public ActionResult Details(string id)
    {
        var category = context.CategoryAds.Find(id);
        if (category == null)
        {
            return HttpNotFound();
        }
        return View(category);
    }
}
```

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

- Title Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), OnlineAdz.
- Solution Explorer:** Shows the project structure for "Solution OnlineAdz" (1 of 1 project). It includes files like AccountController.cs, CategoryAdController.cs, HomeController.cs, PlatformController.cs, UserController.cs, UserInterestController.cs, Migrations, Models, Views, appsettings.json, ApplicationDbContext.cs, Program.cs, and Startup.cs.
- Code Editor:** The current file is "CategoryAdController.cs". The code handles the deletion of a category ad from the database. It includes methods for deleting by ID and by ID with a category ad object.
- Status Bar:** Shows the date and time (04-05-2022) and system status (AWL -5.00%).

## Controllers:HomeController:

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

- Title Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), OnlineAdz.
- Solution Explorer:** Shows the project structure for "Solution OnlineAdz" (1 of 1 project). It includes files like AccountController.cs, CategoryAdController.cs, HomeController.cs, PlatformController.cs, UserController.cs, UserInterestController.cs, Migrations, Models, Views, appsettings.json, ApplicationDbContext.cs, Program.cs, and Startup.cs.
- Code Editor:** The current file is "HomeController.cs". It defines a controller that implements the Controller interface. It has three actions: Index, Privacy, and AboutUs, each returning a View.
- Status Bar:** Shows the date and time (04-05-2022) and system status (28°C Light rain).

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

- MenuBar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- Toolbars:** Standard toolbar with icons for file operations, search, and navigation.
- Code Editor:** The active tab is "HomeController.cs" under the "OnlineAdz" project. The code defines several action methods: Index, Privacy, AboutUs, ContactUs, and Error. The Error method uses ResponseCache and returns a new ErrorViewModel.
- Solution Explorer:** Shows the project structure for "Solution OnlineAdz" (1 of 1 project) containing "Controllers", "Models", and "Views" folders, along with files like AccountController.cs, CategoryAdController.cs, HomeController.cs, PlatformController.cs, UserAdController.cs, UserInterestController.cs, appsettings.json, Program.cs, and Startup.cs.
- Status Bar:** Shows the date (04-05-2022), time (12:34), and weather (28°C Light rain).

## Controllers:PlatformController:

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

- MenuBar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- Toolbars:** Standard toolbar with icons for file operations, search, and navigation.
- Code Editor:** The active tab is "PlatformController.cs" under the "OnlineAdz" project. The code defines a single action method "Index" which returns a view of the platforms.
- Solution Explorer:** Shows the project structure for "Solution OnlineAdz" (1 of 1 project) containing "Controllers", "Models", and "Views" folders, along with files like AccountController.cs, CategoryAdController.cs, HomeController.cs, PlatformController.cs, UserAdController.cs, UserInterestController.cs, appsettings.json, Program.cs, and Startup.cs.
- Status Bar:** Shows the date (04-05-2022), time (12:34), and weather (28°C Light rain).

## Controllers:UserAdController:

The screenshot shows the Visual Studio IDE interface with the UserAdController.cs file open in the main editor. The code implements a controller for managing user ads. It includes methods for indexing ads by email or ad name, creating new ads, and displaying individual ads. The Solution Explorer on the right shows the project structure with various controllers like AccountController, CategoryAdController, HomeController, PlatformController, UserAdController, and UserInterestController.

```
using Microsoft.AspNetCore.Mvc;
using OnlineAdz.Models;

namespace OnlineAdz.Controllers
{
    [Authorize]
    public class UserAdController : Controller
    {
        private readonly AdApplicationContext context = new AdApplicationContext();

        public ActionResult Index(string option, string search)
        {
            if (option == "Email")
            {
                return View(context.UserAds.Where(x => x.Email == search).ToList());
            }
            else
            {
                return View(context.UserAds.Where(x => x.AdName == search).ToList());
            }
        }

        public IActionResult Create()
        {
            return View();
        }
    }
}
```

The screenshot shows the Visual Studio IDE interface with the UserAdController.cs file open in the main editor. The code now includes additional methods: Create, Edit, and Edit. The Create method adds a new ad to the database and redirects to the index. The Edit and Edit methods update existing ads in the database. The Solution Explorer on the right shows the project structure with various controllers like AccountController, CategoryAdController, HomeController, PlatformController, UserAdController, and UserInterestController.

```
        return View();
    }

    [HttpPost]
    public IActionResult Create(UserAd ad)
    {
        if (ad != null)
        {
            context.UserAds.Add(ad);
            context.SaveChanges();
            return RedirectToAction("Index");
        }
        return View();
    }

    public IActionResult Edit(int id)
    {
        var ad = context.UserAds.Find(id);
        return View(ad);
    }

    [HttpPost]
    public IActionResult Edit(int id, UserAd ad)
    {
        if (ad != null)
        {
            UserAd a = context.UserAds.Find(id);
            a.CategoryId = ad.CategoryId;
        }
    }
}
```

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `UserAdController.cs` file under the `OnlineAdz` project. The code implements the `IActionResult Delete` method, which finds a user ad by ID, removes it from the context, and saves changes. It also contains a `Details` method that returns a view for a specific user ad. The Solution Explorer on the right shows the project structure with various controllers like AccountController, CategoryAdController, HomeController, PlatformController, UserAdController, and UserInterestController.

```
if (ad != null)
{
    UserAd a = context.UserAds.Find(id);
    a.CategoryId = ad.CategoryId;
    a.Email = ad.Email;
    a.AdName = ad.AdName;
    a.AdStatus = ad.AdStatus;
    a.UserInterest = ad.UserInterest;
    context.SaveChanges();
    return RedirectToAction("Index");
}

return View();
}

[HttpPost]
[ValidateAntiForgeryToken]
public IActionResult Delete(int id, UserAd ad)
{
    if (ad == null)
    {
        UserAd a = context.UserAds.Find(id);
        context.UserAds.Remove(a);
        context.SaveChanges();
        return RedirectToAction("Index");
    }
}
```

This screenshot shows the same `UserAdController.cs` file with additional methods: `Index` and `Details`. The `Index` method returns a view for all user ads. The `Details` method takes an integer parameter `id` and returns a view for a specific user ad. The rest of the code remains the same as in the previous screenshot.

```
return View(context.UserAds.Find(id));
}

[HttpPost]
[ValidateAntiForgeryToken]
public IActionResult Delete(int id, UserAd ad)
{
    if (ad == null)
    {
        UserAd a = context.UserAds.Find(id);
        context.UserAds.Remove(a);
        context.SaveChanges();
        return RedirectToAction("Index");
    }
    return View();
}

[HttpPost]
[ValidateAntiForgeryToken]
public IActionResult Details(int id)
{
    return View(context.UserAds.Find(id));
}
```

## Controllers:UserInterestController:

The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and a search bar for 'Search (Ctrl+Q)'. The title bar displays 'OnlineAdz' and 'OnlineAdz'. The left sidebar has 'Server Explorer' and 'Toolbox' tabs. The main editor window shows the 'UserInterestController.cs' file:

```
using Microsoft.AspNetCore.Mvc;
using OnlineAdz.Models;
namespace OnlineAdz.Controllers
{
    [ApiController]
    public class UserInterestController : Controller
    {
        [HttpGet]
        public IActionResult Index()
        {
            return View();
        }
        [HttpPost]
        public IActionResult Index(UserInterest ui)
        {
            string email = ui.Email;
            string categoryId = ui.CategoryId;
            string adname = ui.AdName;
            string addesc = ui.AdDesc;
            return View();
        }
    }
}
```

The Solution Explorer on the right lists the project 'OnlineAdz' with its files:

- Connected Services
- Dependencies
- Properties
- wwwroot
- Controllers
  - CategoryController.cs
  - CategoryAdminController.cs
  - HomeController.cs
  - PlatformController.cs
  - UserAdminController.cs
  - UserInterestController.cs
- Migrations
- Models
- Views
- appsettings.json
- DataContext.cs
- Program.cs
- Startup.cs

The bottom navigation bar includes tabs for Error List, Breakpoints, Immediate Window, Output, and a 'Ready' status indicator. The status bar at the bottom right shows the date as 04-05-2022, the time as 12:37, and the weather as 28°C Light rain.

### Models: AdApplicationContext:

The screenshot shows the Microsoft Visual Studio IDE interface. The main area displays the code for `AdApplicationContext.cs` in the `OnlineAdz` project. The code defines a `DbContext` class named `AdApplicationContext` that inherits from `DbContext`. It includes properties for `CategoryAds` and `UserAds`, and a protected override method for `OnConfiguring`. The Solution Explorer window on the right shows the project structure with files like `CategoryAd.cs`, `ErrnViewModel.cs`, and `Startup.cs`. The Properties window is also visible.

```
using System;
using System.Collections.Generic;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Metadata;

namespace OnlineAdz.Models
{
    public partial class AdApplicationContext : DbContext
    {
        public AdApplicationContext()
        {
        }

        public AdApplicationContext(DbContextOptions<AdApplicationContext> options)
            : base(options)
        {
        }

        public virtual DbSet<CategoryAd> CategoryAds { get; set; } = null!;
        public virtual DbSet<UserAd> UserAds { get; set; } = null!;

        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            if (!optionsBuilder.IsConfigured)
            {
                #warning To protect potentially sensitive information in your connection string, you should move it out of source code. You can avoid scaffolding the c
                optionsBuilder.UseSqlServer("Server=;Initial Catalog=AdApplication;User Id=sa;Password=12345");
            }
        }
    }
}
```

The screenshot shows the Microsoft Visual Studio IDE interface. The main area displays the code for `AdApplicationContext.cs` in the `OnlineAdz` project. The code defines two methods: `OnConfiguring` and `OnModelCreating`. The `OnModelCreating` method configures the Entity Framework to use a SQL Server database with specific connection details. The `OnModelCreating` method also applies entity-level configurations, such as setting primary keys and specifying maxLength for properties.

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
{
    if (!optionsBuilder.IsConfigured)
    {
        #warning To protect potentially sensitive information in your connection string, you should move it out of source code. You can avoid scaffolding the OnConfiguring method by using the 'EntityTypeConfiguration' class. For more guidance on开朗配置，请参阅 https://go.microsoft.com/fwlink/?linkid=808683
        optionsBuilder.UseSqlServer("Server=.;Initial Catalog=AdApplication;User Id=sa;Password=l23h5#");
    }
}

protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity<CategoryAd>(entity =>
    {
        entity.HasKey(e => e.CategoryId)
            .HasName("PK__Category__19093A8BF1942A85");

        entity.Property(e => e.CategoryId)
            .HasMaxLength(50)
            .IsUnicode(false);

        entity.Property(e => e.CategoryName)
            .HasMaxLength(50)
            .IsUnicode(false);
    });

    modelBuilder.Entity<UserId>(entity =>
    {
        entity.HasKey(e => e.Id)
            .HasName("PK__UserAdz__7138D5AEB7D9A2E9");
    });
}
```

The Solution Explorer on the right shows the project structure, including files like `CategoryAd.cs`, `ErrorViewModel.cs`, and `Startup.cs`. The bottom right corner displays the date and time as "04-05-2022 12:38".

The screenshot shows a Microsoft Visual Studio interface with the following components:

- Code Editor:** Displays the `AdApplicationContext.cs` file containing C# code for Entity Framework configuration.
- Solution Explorer:** Shows the project structure for "OnlineAdz" with files like `CategoryAd.cs`, `ErrorViewModel.cs`, and `Startup.cs`.
- Output Window:** Shows the message "Show output from: Service Dependencies".

## Models:CategoryAd:

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

- Title Bar:** OnlineAdz
- Toolbars:** Standard, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Code Editor:** CategoryAd.cs (OnlineAdz.Models) - Shows the definition of the CategoryAd class with properties CategoryId, CategoryName, and UserAds.
- Solution Explorer:** Shows the project structure for "Solution OnlineAdz" (1 of 1 project) with files like AdApplicationContext.cs, CategoryAd.cs, ErrorViewModel.cs, LoginViewModel.cs, Platform.cs, popularityContext.cs, RegisterViewModel.cs, UserAd.cs, and UserInterests.cs.
- Status Bar:** Ready, Type here to search, 28°C Light rain, 12:39, ENG, 04-05-2022.

## Models:ErrorViewModel:

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

- Title Bar:** OnlineAdz
- Toolbars:** Standard, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Code Editor:** ErrorViewModel.cs (OnlineAdz.Models) - Shows the definition of the ErrorViewModel class with a RequestId property and a ShowRequestId method.
- Solution Explorer:** Shows the project structure for "Solution OnlineAdz" (1 of 1 project) with files like AdApplicationContext.cs, CategoryAd.cs, ErrorViewModel.cs, LoginViewModel.cs, Platform.cs, popularityContext.cs, RegisterViewModel.cs, UserAd.cs, and UserInterests.cs.
- Status Bar:** Ready, Type here to search, 28°C Light rain, 12:39, ENG, 04-05-2022.

## Models:LoginViewModel:

```
namespace OnlineAdz.Models
{
    public class LoginViewModel
    {
        public string Email { get; set; }
        [DataType(DataType.Password)]
        public string Password { get; set; }
        public string Role { get; set; }
    }
}
```

## Models:Platform:

```
using System;
using System.Collections.Generic;
namespace OnlineAdz.Models
{
    public partial class Platform
    {
        public string CategoryId { get; set; } = null!;
        public string CategoryName { get; set; } = null!;
        public string AddressDescription { get; set; } = null!;
        public string MainPlatform { get; set; } = null!;
    }
}
```

## Models:popularityContext:

```
using System;
using System.Collections.Generic;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Metadata;

namespace OnlineAdz.Models
{
    public partial class popularityContext : DbContext
    {
        public popularityContext()
        {
        }

        public popularityContext(DbContextOptions<popularityContext> options)
            : base(options)
        {
        }

        public virtual DbSet<Platform> Platforms { get; set; } = null!;

        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            if (!optionsBuilder.IsConfigured)
            {
                #warning To protect potentially sensitive information in your connection string, you should move it out of source code. You can avoid scaffolding the c
                optionsBuilder.UseSqlServer("Server=.;Initial Catalog=popularity;User Id=sa;Password=12345");
            }
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Metadata;

namespace OnlineAdz.Models
{
    public partial class popularityContext : DbContext
    {
        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            modelBuilder.Entity<Platform>(entity =>
            {
                entity.HasKey();

                entity.Property(e => e.AdDescription)
                    .HasMaxLength(90)
                    .IsUnicode(false);

                entity.Property(e => e.CategoryId)
                    .HasMaxLength(50)
                    .IsUnicode(false);

                entity.Property(e => e.CategoryName)
                    .HasMaxLength(50)
                    .IsUnicode(false);

                entity.Property(e => e.MainPlatform)
                    .HasMaxLength(50)
                    .IsUnicode(false);
            });

            OnModelCreatingPartial(modelBuilder);
        }

        protected override void OnModelCreatingPartial(ModelBuilder modelBuilder)
        {
        }
    }
}
```

## Models: RegisterViewModel:

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

- Title Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), OnlineAdz.
- Solution Explorer:** Shows the project structure for "Solution OnlineAdz" (1 of 1 project). It includes files like Connected Services, Dependencies, Properties, wwwroot, Controllers, Migrations, Models (with RegisterViewModel.cs selected), Views, appsettings.json, ApplicationDbContext.cs, CategoryAd.cs, ErrorViewModel.cs, LoginViewModel.cs, Platform.cs, popularityContext.cs, UserAd.cs, UserInterest.cs, and Startup.cs.
- Code Editor:** The "RegisterViewModel.cs" file is open. The code defines a class "RegisterViewModel" with properties: Name, Email, Role, State, City, and Password, each annotated with [Required] and [EmailAddress] attributes.
- Status Bar:** Shows the date (04-05-2022) and time (12:41).

## Models:UserAd:

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

- Title Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), OnlineAdz.
- Solution Explorer:** Shows the project structure for "Solution OnlineAdz" (1 of 1 project). It includes files like Connected Services, Dependencies, Properties, wwwroot, Controllers, Migrations, Models (with UserAd.cs selected), Views, appsettings.json, ApplicationDbContext.cs, CategoryAd.cs, ErrorViewModel.cs, LoginViewModel.cs, Platform.cs, popularityContext.cs, RegisterViewModel.cs, UserAd.cs, UserInterest.cs, and Startup.cs.
- Code Editor:** The "UserAd.cs" file is open. It contains a partial class "UserAd" with properties: AdId, CategoryId, Email, AdName, AdStatus, and UserInterest, each annotated with [Required] and [EmailAddress] attributes.
- Status Bar:** Shows the date (04-05-2022) and time (12:41).

## Models:UserInterest:

```

namespace OnlineAdz.Models
{
    public class UserInterest
    {
        public string Email { get; set; }
        public string CategoryId { get; set; }
        public string AdName { get; set; }
        public string AdDesc { get; set; }
    }
}

```

## Views: Account:

### Login:

```

<body style="background-image: url('../css/images/fiveresize.jpg'); ">

@model OnlineAdz.Models.LoginViewModel

<h1>Login</h1>
<hr />


<div class="col-md-4">
        <form asp-action="Login" asp-controller="Account">
            <div asp-validation-summary="All" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="Email" class="control-label"></label>
                <input asp-for="Email" class="form-control" placeholder="Email" required/>
                <span asp-validation-for="Email" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Password" class="control-label"></label>
                <input asp-for="Password" class="form-control" placeholder="Password" required/>
                <span asp-validation-for="Password" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Role" class="control-label"></label>
                <input asp-for="Role" class="form-control" placeholder="Role" required/>
                <span asp-validation-for="Role" class="text-danger"></span>
            </div>
            <br />
            <div class="form-group">
                <input type="submit" value="Login" class="btn btn-primary" />
            </div>
        </form>
    </div>


```

### Register:

The screenshot shows the Visual Studio IDE interface. The main window displays the `Register.cshtml` view, which contains form fields for Name, Email, Password, Role, State, and City. The `Solution Explorer` pane on the right shows the project structure for `OnlineAdz`, including files like `Login.cshtml`, `Register.cshtml`, `CategoryAd.cshtml`, and `Startup.cs`. The status bar at the bottom indicates it's 28°C Light rain, 12:44, and the date is 04-05-2022.

```
<body style="background-image:url('../Images/fiveresize.jpg');">
    <div class="row">
        <div class="col-md-4">
            <form asp-action="Register" asp-controller="Account">
                <div asp-validation-summary="All" class="text-danger"></div>
                <div class="form-group">
                    <label asp-for="Name" class="control-label"></label>
                    <input asp-for="Name" class="form-control" placeholder="Name"/>
                    <span asp-validation-for="Name" class="text-danger"></span>
                </div>
                <div class="form-group">
                    <label asp-for="Email" class="control-label"></label>
                    <input asp-for="Email" class="form-control" placeholder="Email"/>
                    <span asp-validation-for="Email" class="text-danger"></span>
                </div>
                <div class="form-group">
                    <label asp-for="Password" class="control-label"></label>
                    <input asp-for="Password" class="form-control" placeholder="Password" />
                    <span asp-validation-for="Password" class="text-danger"></span>
                </div>
                <div class="form-group">
                    <label asp-for="Role" class="control-label"></label>
                    <input asp-for="Role" class="form-control" placeholder="Role"/>
                    <span asp-validation-for="Role" class="text-danger"></span>
                </div>
                <div class="form-group">
                    <label asp-for="State" class="control-label"></label>
                    <input asp-for="State" class="form-control" placeholder="State"/>
                    <span asp-validation-for="State" class="text-danger"></span>
                </div>
                <div class="form-group">
                    <label asp-for="City" class="control-label"></label>
                    <input asp-for="City" class="form-control" placeholder="City"/>
                    <span asp-validation-for="City" class="text-danger"></span>
                </div>
                <br />
                <div class="form-group">
                    <input type="submit" value="Register" class="btn btn-primary" />
                </div>
            </form>
        </div>
    </div>
    <div asp-validation-summary="ValidationScriptsPartial" class="text-danger" style="font-size: small; margin-top: 10px;">
```

This screenshot is nearly identical to the one above, showing the `Register.cshtml` view and the `OnlineAdz` project in the `Solution Explorer`. The status bar at the bottom indicates it's 28°C Light rain, 12:44, and the date is 04-05-2022.

```
<body style="background-image:url('../Images/fiveresize.jpg');">
    <div class="row">
        <div class="col-md-4">
            <form asp-action="Register" asp-controller="Account">
                <div asp-validation-summary="All" class="text-danger"></div>
                <div class="form-group">
                    <label asp-for="Password" class="control-label"></label>
                    <input asp-for="Password" class="form-control" placeholder="Password" />
                    <span asp-validation-for="Password" class="text-danger"></span>
                </div>
                <div class="form-group">
                    <label asp-for="Role" class="control-label"></label>
                    <input asp-for="Role" class="form-control" placeholder="Role"/>
                    <span asp-validation-for="Role" class="text-danger"></span>
                </div>
                <div class="form-group">
                    <label asp-for="State" class="control-label"></label>
                    <input asp-for="State" class="form-control" placeholder="State"/>
                    <span asp-validation-for="State" class="text-danger"></span>
                </div>
                <div class="form-group">
                    <label asp-for="City" class="control-label"></label>
                    <input asp-for="City" class="form-control" placeholder="City"/>
                    <span asp-validation-for="City" class="text-danger"></span>
                </div>
                <br />
                <div class="form-group">
                    <input type="submit" value="Register" class="btn btn-primary" />
                </div>
            </form>
        </div>
    </div>
    <div asp-validation-summary="ValidationScriptsPartial" class="text-danger" style="font-size: small; margin-top: 10px;">
```

## Views: CategoryAd:

ViewStart:

```
Layout = "~/views/Shared/_Layout1.cshtml";
```

## Create:

The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and Search (Ctrl+Q). The title bar displays "OnlineAdz". The left sidebar has "Server Explorer" and "Toolbox" tabs. The main code editor window contains C# code for a "Create.cshtml" view, which is part of the "CategoryAd" controller. The code uses Bootstrap classes like "form-group" and "form-control" for form fields. The Solution Explorer on the right shows the project structure for "OnlineAdz", including files like "CategoryAd.cs", "CategoryAdController.cs", "CategoryAd.cshtml", and "Index.cshtml". The bottom status bar shows "Ready", "Error List", "Breakpoints", "Immediate Window", "Output", "Add to Source Control", "Select Repository", "28°C Light rain", "1246", "ENG", and "04-05-2022".

```
@model OnlineAdz.Models.CategoryAd

@{
    Layout = "_Layout";
    ViewData["Title"] = "Create";
}



# Add new category



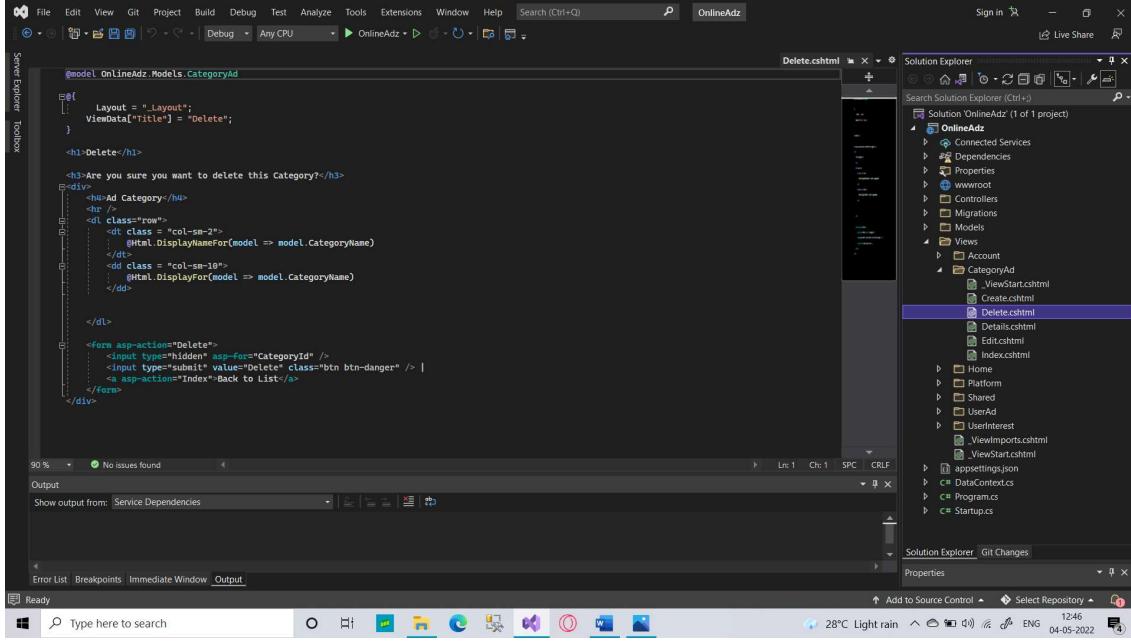
## Ad Categories



---


```

Delete:



The screenshot shows the Visual Studio IDE interface with the 'Delete.cshtml' view file open in the main editor. The code is as follows:

```
@@@
{
    Layout = "_Layout";
    ViewData["Title"] = "Delete";
}



### Delete



Are you sure you want to delete this Category?



Category



|                                                   |                                               |
|---------------------------------------------------|-----------------------------------------------|
| @Html.DisplayNameFor(model => model.CategoryName) | @Html.DisplayFor(model => model.CategoryName) |
|---------------------------------------------------|-----------------------------------------------|

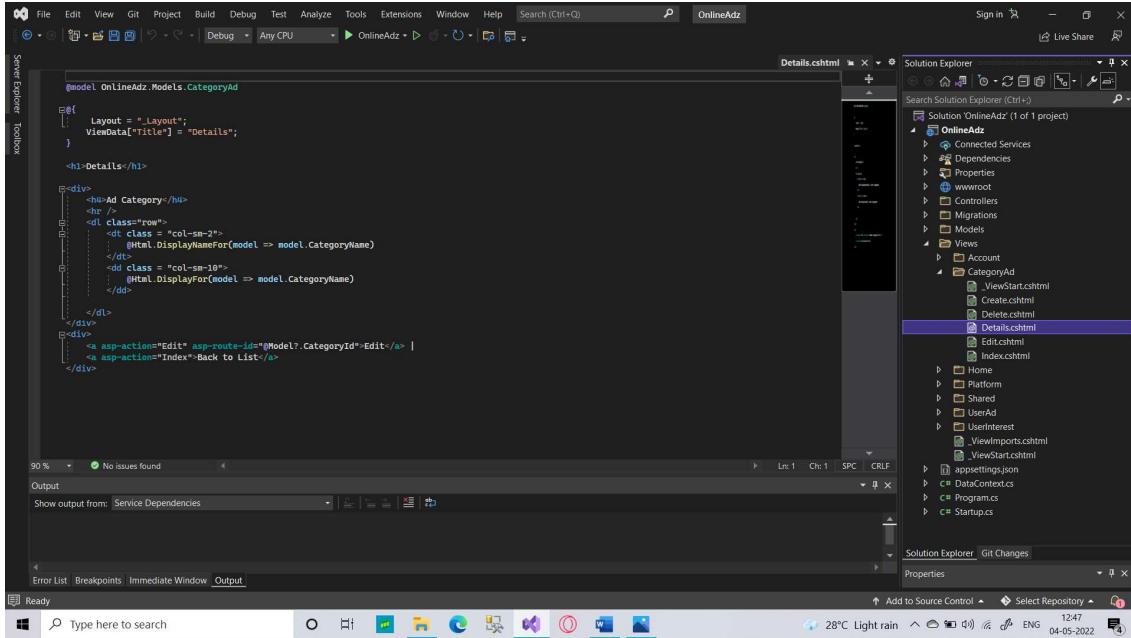


>
        >
        Back to List


```

The Solution Explorer on the right shows the project structure, including the 'Views/CategoryAd/Delete.cshtml' file selected.

Details:



The screenshot shows the Visual Studio IDE interface with the 'Details.cshtml' view file open in the main editor. The code is as follows:

```
@@@
{
    Layout = "_Layout";
    ViewData["Title"] = "Details";
}



### Details



Category
    

|                                                   |                                               |
|---------------------------------------------------|-----------------------------------------------|
| @Html.DisplayNameFor(model => model.CategoryName) | @Html.DisplayFor(model => model.CategoryName) |
|---------------------------------------------------|-----------------------------------------------|

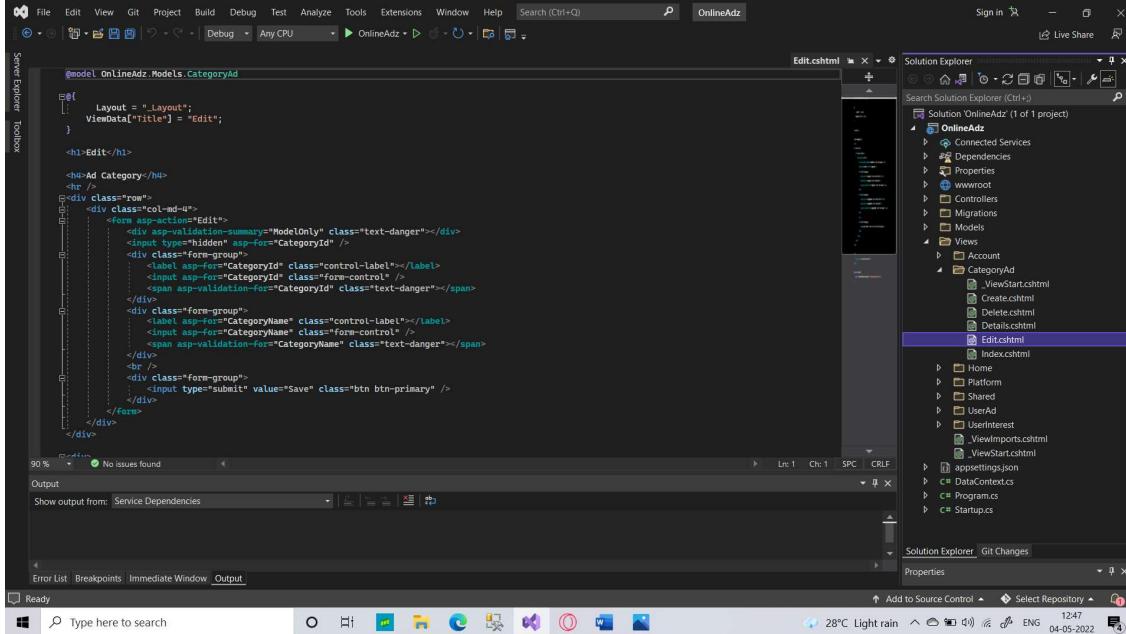


Edit
Index


```

The Solution Explorer on the right shows the project structure, including the 'Views/CategoryAd/Details.cshtml' file selected.

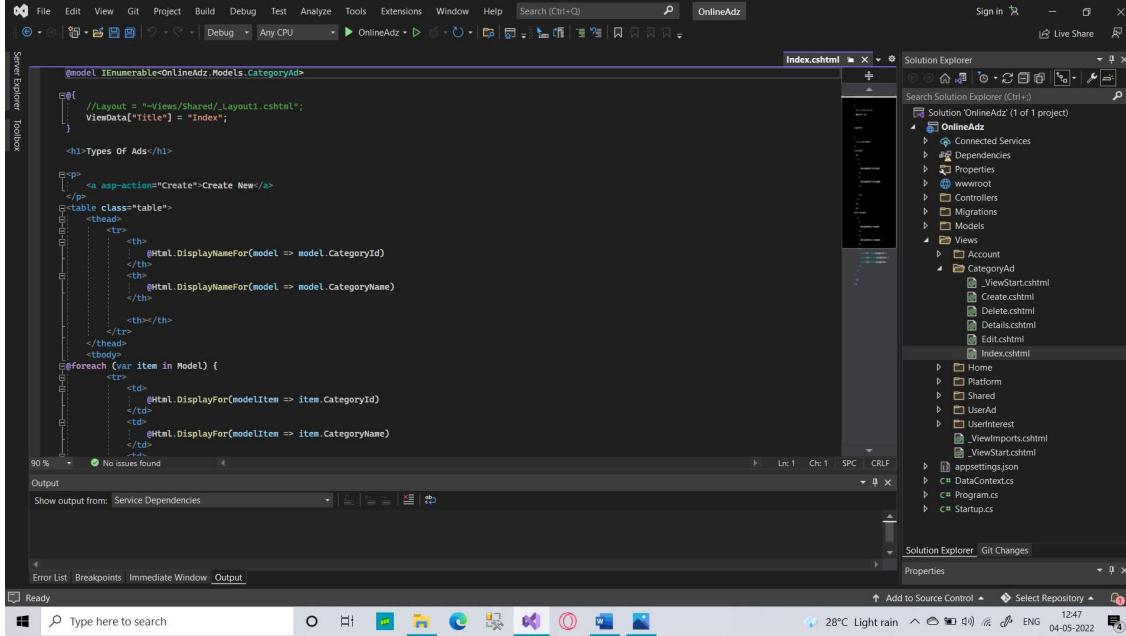
## Edit:



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

- Title Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), OnlineAdz.
- Solution Explorer:** Shows the solution structure for "OnlineAdz" with files like Index.cshtml, Edit.cshtml, Details.cshtml, Create.cshtml, Delete.cshtml, ViewStart.cshtml, and CategoryAd.cs.
- Code Editor:** Displays the "Edit.cshtml" file content, which is a form for editing a category. It includes fields for CategoryId and CategoryName, and a submit button labeled "Save".
- Status Bar:** Shows "Ready", "Type here to search", and system information like "28°C Light rain", "ENG 12:47 04-05-2022".

## Index:



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

- Title Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), OnlineAdz.
- Solution Explorer:** Shows the solution structure for "OnlineAdz" with files like Index.cshtml, Edit.cshtml, Details.cshtml, Create.cshtml, Delete.cshtml, ViewStart.cshtml, and CategoryAd.cs.
- Code Editor:** Displays the "Index.cshtml" file content, which lists categories in a table. It includes a header row with "Types of Ads" and a body row with "Create New".
- Status Bar:** Shows "Ready", "Type here to search", and system information like "28°C Light rain", "ENG 12:47 04-05-2022".

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `Index.cshtml` file content:

```

<thead>
    <tr>
        <th>@Html.DisplayNameFor(model => model.CategoryId)</th>
        <th>@Html.DisplayNameFor(model => model.CategoryName)</th>
    </tr>
</thead>
<tbody>
    @foreach (var item in Model) {
        <tr>
            <td>@Html.DisplayFor(modelItem => item.CategoryId)</td>
            <td>@Html.DisplayFor(modelItem => item.CategoryName)</td>
            <td>
                <a href="#" asp-action="Edit" asp-route-id="@item.CategoryId">Edit</a> | 
                <a href="#" asp-action="Details" asp-route-id="@item.CategoryId">Details</a> | 
                <a href="#" asp-action="Delete" asp-route-id="@item.CategoryId">Delete</a>
            </td>
        </tr>
    }
</tbody>

```

The Solution Explorer on the right shows the project structure for "OnlineAdz" with files like `Index.cshtml`, `CategoryAdController.cs`, and `Startup.cs`.

## Views: Home:

### AboutUs:

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `AboutUs.cshtml` file content:

```

@{
    ViewData["Title"] = "About Us";
}
<h1>@ViewData["Title"]<br /></h1><br />
<p>OnlineAdz is an online advertising portal. Choose from various categories and post your Ad.<br />We help you to build up your brand's name.</p>

```

The Solution Explorer on the right shows the project structure for "OnlineAdz" with files like `AboutUs.cshtml`, `ContactUs.cshtml`, `Index.cshtml`, and `Privacy.cshtml`.

## ContactUs:

The screenshot shows the Visual Studio IDE interface with the 'ContactUs.cshtml' file open in the main editor. The code displays basic HTML structure including a title, phone number, and email address.

```
<@{ ViewData["Title"] = "Contact Us"; }>
<h1><u>@ViewData["Title"]</u><br /><br /></h1>
<p><h3>Phone: </h3><u>123456789</u><br /></p>
<p><h3>Email: </h3><u>admin@OnlineAdz.com</u><br /></p>
```

The Solution Explorer on the right shows the project structure for 'OnlineAdz' with files like 'Index.cshtml', 'AboutUs.cshtml', and 'Privacy.cshtml' under the 'Views/Home' folder.

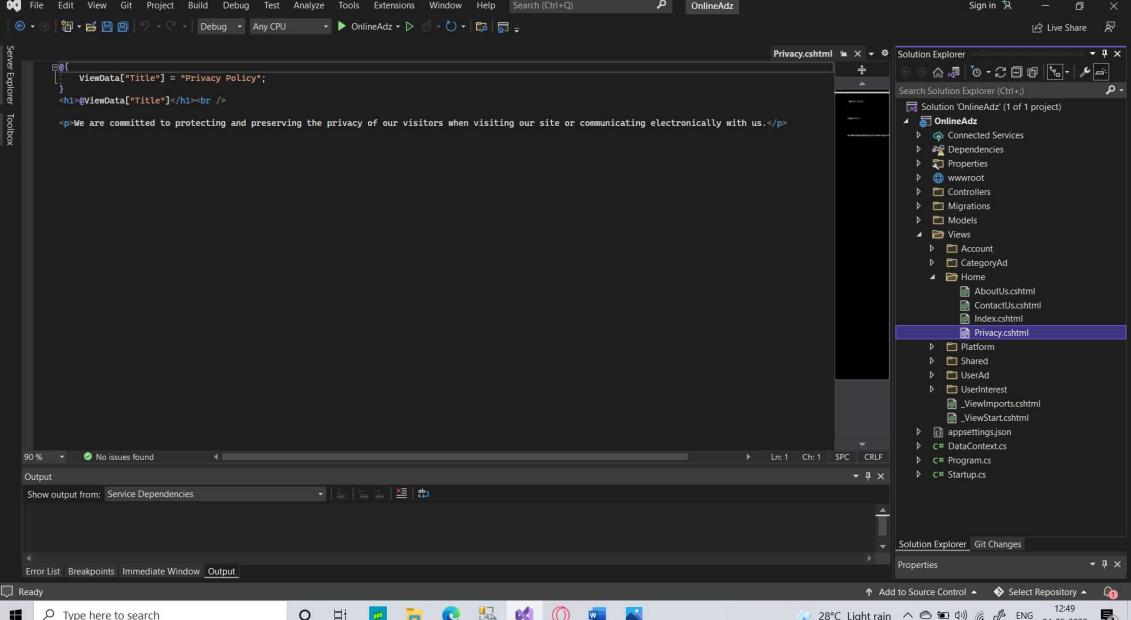
## Index:

The screenshot shows the Visual Studio IDE interface with the 'Index.cshtml' file open in the main editor. The code includes a welcome message and an image placeholder.

```
<@{ ViewData["Title"] = "Home Page"; }>
<div class="text-center">
    <h1><u>Welcome to OnlineAdz!</u><br /><u>We make advertising easy!!</u></h1><br />
    
</div>
```

The Solution Explorer on the right shows the project structure for 'OnlineAdz' with files like 'Index.cshtml', 'AboutUs.cshtml', and 'Privacy.cshtml' under the 'Views/Home' folder.

## Privacy:



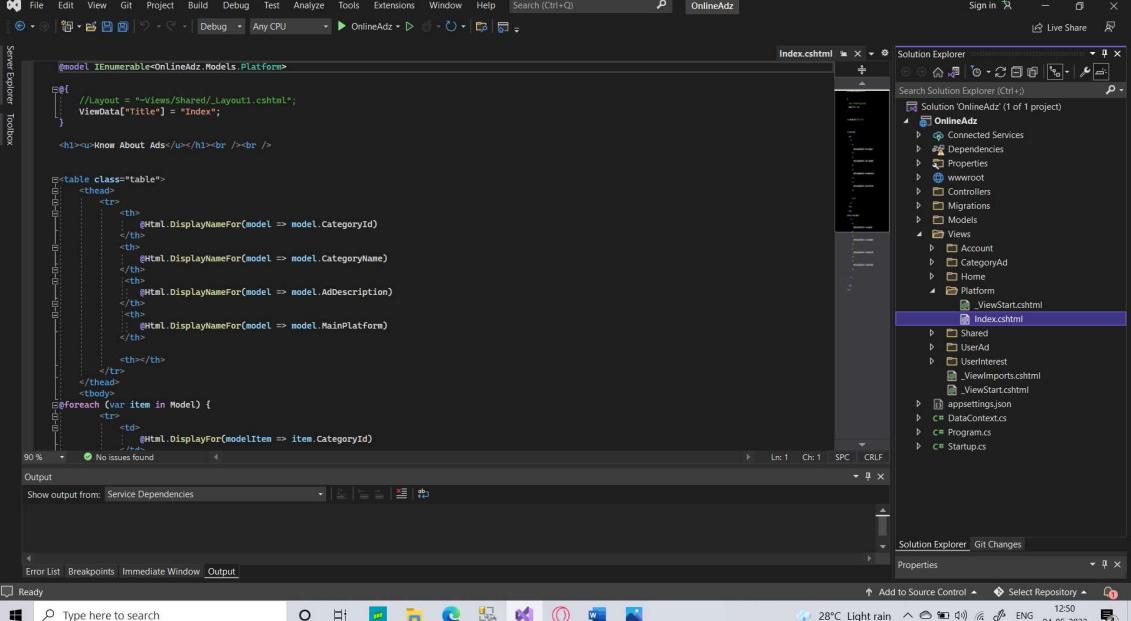
The screenshot shows the Visual Studio IDE interface with the 'Privacy.cshtml' file open in the main editor. The code contains the following HTML and C# code:

```
@{
    ViewData["Title"] = "Privacy Policy";
}
<h1>@ViewData["Title"]</h1><br />
<p>We are committed to protecting and preserving the privacy of our visitors when visiting our site or communicating electronically with us.</p>
```

The Solution Explorer on the right shows the project structure for 'OnlineAdz' with files like 'Index.cshtml', 'Privacy.cshtml', and 'Startup.cs'.

## Views: Platform:

## Index:



The screenshot shows the Visual Studio IDE interface with the 'Index.cshtml' file open in the main editor. The code contains the following HTML and C# code:

```
@model IEnumerable<OnlineAdz.Models.Platform>
@{
    //Layout = "~/Views/Shared/_Layout1.cshtml";
    ViewData["Title"] = "Index";
}

<h1>@<a href="#">Know About Ads</a></h1><br /><br />

<table class="table">
    <thead>
        <tr>
            <th>@Html.DisplayNameFor(model => model.CategoryId)</th>
            <th>@Html.DisplayNameFor(model => model.CategoryName)</th>
            <th>@Html.DisplayNameFor(model => model.AdDescription)</th>
            <th>@Html.DisplayNameFor(model => model.MainPlatform)</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model) {
            <tr>
                <td>@Html.DisplayFor(modelItem => item.CategoryId)</td>
                <td>@Html.DisplayFor(modelItem => item.CategoryName)</td>
                <td>@Html.DisplayFor(modelItem => item.AdDescription)</td>
                <td>@Html.DisplayFor(modelItem => item.MainPlatform)</td>
            </tr>
        }
    </tbody>
</table>
```

The Solution Explorer on the right shows the project structure for 'OnlineAdz' with files like 'Index.cshtml', 'Privacy.cshtml', and 'Startup.cs'.

The screenshot shows the Visual Studio IDE interface. The main window displays the code for `Index.cshtml`, which contains a table structure with multiple columns and rows. The Solution Explorer on the right shows the project structure for "OnlineAdz" with files like `Index.cshtml`, `Startup.cs`, and `Program.cs`. The status bar at the bottom indicates the date and time as 04-05-2022.

```

<tbody>
    <tr>
        <td>@Html.DisplayNameFor(model => model.MainPlatform)</td>
        <td></td>
    </tr>
</tbody>

```

## Views: Shared:

### \_Layout:

The screenshot shows the Visual Studio IDE interface. The main window displays the code for `_Layout.cshtml`, which includes a header section with a navigation bar and a body section. The Solution Explorer on the right shows the project structure for "OnlineAdz" with files like `_Layout.cshtml`, `Index.cshtml`, and `Startup.cs`. The status bar at the bottom indicates the date and time as 04-05-2022.

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>ViewData["Title"] - OnlineAdz</title>
        <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
        <link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />
        <link rel="stylesheet" href="~/OnlineAdz.styles.css" asp-append-version="true" />
    </head>
    <body>
        <header>
            <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">
                <div class="container-fluid">
                    <a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">OnlineAdz</a>
                    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
                        <span class="navbar-toggler-icon"></span>
                    </button>
                    <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
                        <ul class="navbar-nav flex-grow-1">
                            <li class="nav-item">
                                <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link text-dark" asp-area="" asp-controller="Account" asp-action="Register">Register</a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link text-dark" asp-area="" asp-controller="Account" asp-action="Login">Login</a>
                            </li>
                        </ul>
                    </div>
                </div>
            </nav>
        </header>
        <div class="container">
            <partial name="NavPartial" />
            <main>
                <partial name="ContentPartial" />
            </main>
            <partial name="FooterPartial" />
        </div>
    </body>

```

## \_Layout1:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>OnlineAdz</title>
        <link rel="stylesheet" href="/lib/bootstrap/dist/css/bootstrap.min.css" />
        <link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />
        <link rel="stylesheet" href="~/OnlineAdz.styles.css" asp-append-version="true" />
    </head>
    <body style="background-image:url('../css/images/sixteen.jpg');">
        <header>
            <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">
                <a class="navbar-brand" asp-area="" asp-controller="CategoryAd" asp-action="Index">OnlineAdz</a>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
                    <ul class="navbar-nav flex-grow-1">
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-controller="CategoryAd" asp-action="Index">Ad Category</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-controller="UserAd" asp-action="Index">Users and Ads</a>
                        </li>
                    </ul>
                </div>
            </nav>
        </header>
        <div class="container-fluid">
            <div class="row">
                <div class="col-12" style="text-align: center; margin-top: 20px;">
                    <img alt="Placeholder image for the main content area" style="width: 100%; height: auto; border-radius: 10px; border: 1px solid #ccc; margin-bottom: 10px;">
                    <h1>Welcome to OnlineAdz!</h1>
                    <p>This is the main landing page for our advertising platform. You can manage ads, view user interests, and explore different categories. Stay tuned for more features coming soon!</p>
                </div>
            </div>
        </div>
    </body>

```

## \_Layout2:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>OnlineAdz</title>
        <link rel="stylesheet" href="/lib/bootstrap/dist/css/bootstrap.min.css" />
        <link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />
        <link rel="stylesheet" href="~/OnlineAdz.styles.css" asp-append-version="true" />
    </head>
    <body style="background-image:url('../css/images/nine.jpg');">
        <header>
            <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">
                <a class="navbar-brand" asp-area="" asp-controller="UserAd" asp-action="Index">OnlineAdz</a>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
                    <ul class="navbar-nav flex-grow-1">
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-controller="UserAd" asp-action="Index">Users and Ads</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-controller="Platform" asp-action="Index">About Ads</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-controller="UserInterest" asp-action="Index">User Interest</a>
                        </li>
                    </ul>
                </div>
            </nav>
        </header>
        <div class="container-fluid">
            <div class="row">
                <div class="col-12" style="text-align: center; margin-top: 20px;">
                    <img alt="Placeholder image for the main content area" style="width: 100%; height: auto; border-radius: 10px; border: 1px solid #ccc; margin-bottom: 10px;">
                    <h1>Welcome to OnlineAdz!</h1>
                    <p>This is the main landing page for our advertising platform. You can manage ads, view user interests, and explore different categories. Stay tuned for more features coming soon!</p>
                </div>
            </div>
        </div>
    </body>
```

## Views: UserAd:

Create:

The screenshot shows the Visual Studio IDE interface with the 'Create.cshtml' view for the 'UserAd' model selected in the Solution Explorer. The code editor displays the following HTML and C# code:

```
#model OnlineAdz.Models.UserAd
@{
    Layout = "_Layout";
    ViewData["Title"] = "Create";
}



# Create



---



<form asp-action="Create">
    <div asp-validation-summary="ModelOnly" class="text-danger"></div>
    <div class="form-group">
        <label asp-for="CategoryId" class="control-label"></label>
        <input asp-for="CategoryId" class="form-control" />
        <span asp-validation-for="CategoryId" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="Email" class="control-label"></label>
        <input asp-for="Email" class="form-control" />
        <span asp-validation-for="Email" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="AdName" class="control-label"></label>
        <input asp-for="AdName" class="form-control" />
        <span asp-validation-for="AdName" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="AdStatus" class="control-label"></label>
        <input asp-for="AdStatus" class="form-control" />
        <span asp-validation-for="AdStatus" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="UserInterest" class="control-label"></label>
        <input asp-for="UserInterest" class="form-control" />
        <span asp-validation-for="UserInterest" class="text-danger"></span>
    </div>
    <div class="form-group">
        <input type="submit" value="Create" class="btn btn-primary" />
    </div>
</form>


<a asp-action="Index">Back to List</a>

```

The Solution Explorer on the right shows the project structure with files like 'Create.cshtml', 'Delete.cshtml', 'Details.cshtml', 'Edit.cshtml', and 'Index.cshtml' under the 'Views/UserAd' folder.

The screenshot shows the Visual Studio IDE interface with the 'Create.cshtml' view for the 'UserAd' model selected in the Solution Explorer. The code editor displays the following HTML and C# code, including a partial view for validation scripts:

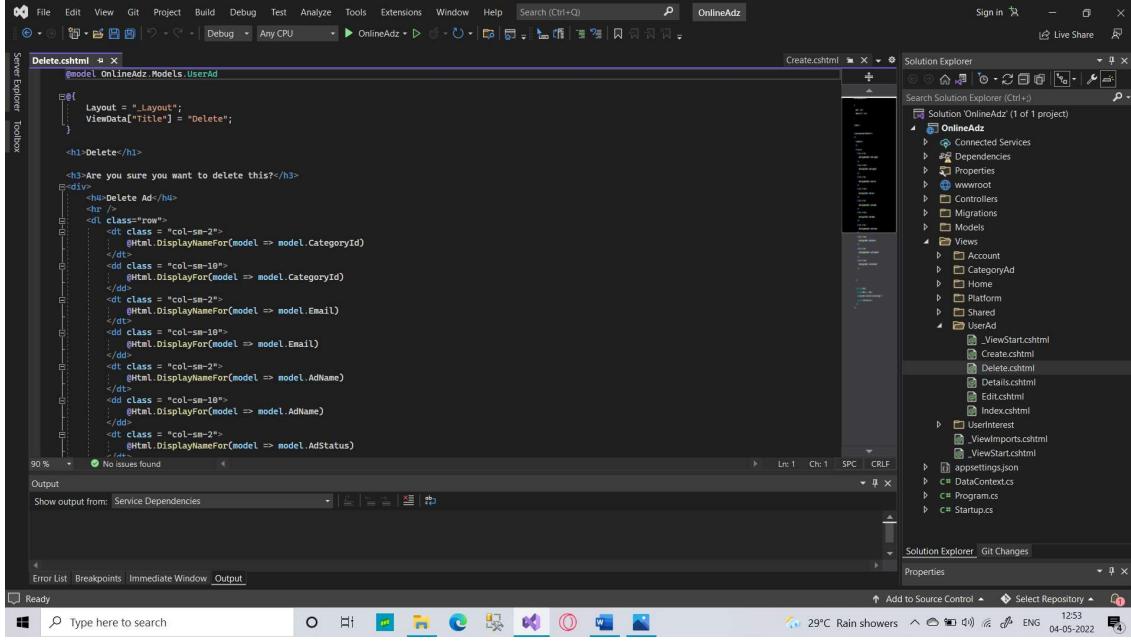
```
<span asp-validation-for="Email" class="text-danger"></span>
</div>
<div class="form-group">
    <label asp-for="AdName" class="control-label"></label>
    <input asp-for="AdName" class="form-control" />
    <span asp-validation-for="AdName" class="text-danger"></span>
</div>
<div class="form-group">
    <label asp-for="AdStatus" class="control-label"></label>
    <input asp-for="AdStatus" class="form-control" />
    <span asp-validation-for="AdStatus" class="text-danger"></span>
</div>
<div class="form-group">
    <label asp-for="UserInterest" class="control-label"></label>
    <input asp-for="UserInterest" class="form-control" />
    <span asp-validation-for="UserInterest" class="text-danger"></span>
</div>
<div class="form-group">
    <input type="submit" value="Create" class="btn btn-primary" />
</div>
</div>
<a asp-action="Index">Back to List</a>

```

```
@section Scripts {
    @await Html.RenderPartialAsync("_ValidationScriptsPartial");
}
```

The Solution Explorer on the right shows the project structure with files like 'Create.cshtml', 'Delete.cshtml', 'Details.cshtml', 'Edit.cshtml', and 'Index.cshtml' under the 'Views/UserAd' folder.

## Delete:

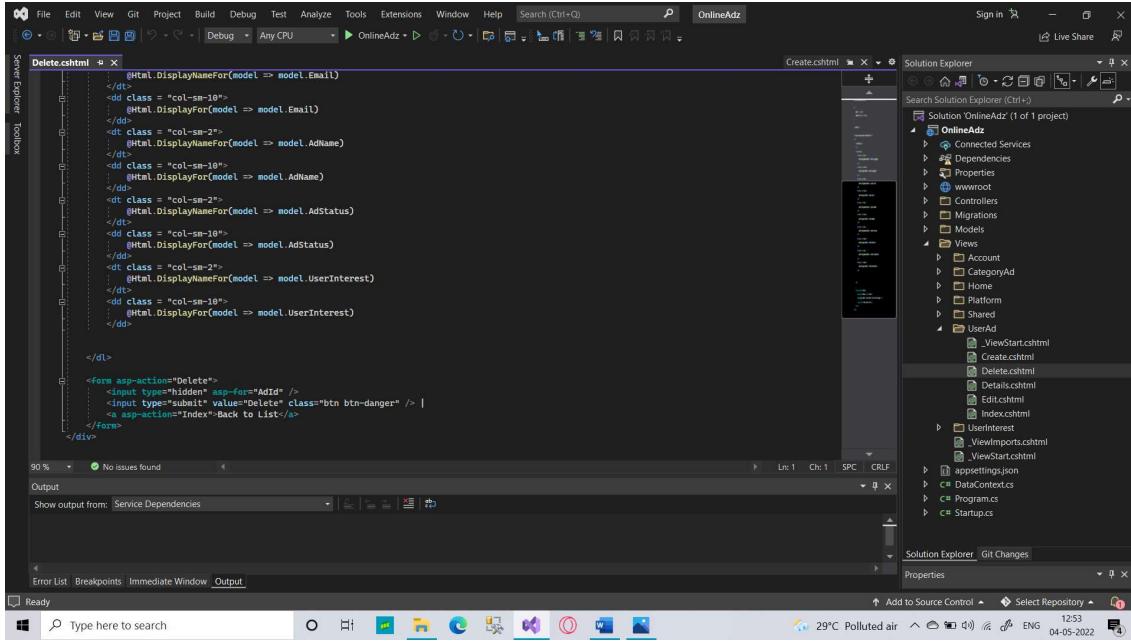


The screenshot shows the Visual Studio IDE interface with the Delete.cshtml file open in the main editor. The code is a standard ASP.NET MVC view for deleting a UserAd. It includes a title, a confirmation message, and a form with a hidden field for the AdId and a submit button labeled "Delete". The Solution Explorer on the right shows the project structure, including the UserAd view folder and its sub-files like ViewStart.cshtml, Create.cshtml, Delete.cshtml, Details.cshtml, Edit.cshtml, and Index.cshtml.

```
<@model OnlineAdz.Models.UserAd>
@{
    Layout = "_Layout";
    ViewData["Title"] = "Delete";
}

<h1>Delete</h1>

<h3>Are you sure you want to delete this?</h3>
<div>
    <p>Delete Ad</p>
    <br />
    <dl class="row">
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.CategoryId)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.CategoryId)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.Email)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.Email)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.AdName)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.AdName)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.AdStatus)
        </dt>
    </dl>
</div>
```



This screenshot shows the same Visual Studio environment, but the Delete.cshtml file now contains additional code. It includes sections for displaying the UserInterest and AdStatus properties of the model. The code is identical to the one in the first screenshot, plus the new sections for UserInterest and AdStatus.

```
<@model OnlineAdz.Models.UserAd>
@{
    Layout = "_Layout";
    ViewData["Title"] = "Delete";
}

<h3>Are you sure you want to delete this?</h3>
<div>
    <p>Delete Ad</p>
    <br />
    <dl class="row">
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.Email)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.Email)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.AdName)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.AdName)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.AdStatus)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.AdStatus)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.UserInterest)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.UserInterest)
        </dd>
    </dl>
    <form asp-action="Delete">
        <input type="hidden" asp-for="AdId" />
        <input type="submit" value="Delete" class="btn btn-danger" /> |
        <a href="#" asp-action="Index">Back to List</a>
    </form>
</div>
```

## Details:

The screenshot shows the Visual Studio IDE interface. The left pane displays the code for `Details.cshtml`, which contains ASP.NET Razor syntax for displaying user details. The right pane shows the `Solution Explorer` with the project structure for `OnlineAdz`. The `Views/UserAd` folder is expanded, showing files like `Index.cshtml`, `Create.cshtml`, `Delete.cshtml`, and `Details.cshtml`. The status bar at the bottom indicates the date as 04-05-2022.

```
<@model OnlineAdz.Models.UserAd>
<div>
    <h1>Details</h1>
    <div>
        <table border="1">
            <tr>
                <td>Email</td>
                <td>Ad Name</td>
            </tr>
            <tr>
                <td><@Html.DisplayFor(model => model.Email)></td>
                <td><@Html.DisplayFor(model => model.AdName)></td>
            </tr>
            <tr>
                <td>Ad Status</td>
                <td><@Html.DisplayFor(model => model.AdStatus)></td>
            </tr>
            <tr>
                <td>User Interest</td>
                <td><@Html.DisplayFor(model => model.UserInterest)></td>
            </tr>
        </table>
    </div>
    <div>
        <a href="#">Edit</a> |
        <a href="#">Index</a> |
        <a href="#">Back to List</a>
    </div>
</div>
```

This screenshot is nearly identical to the one above, showing the same code for `Details.cshtml` and the same project structure in the `Solution Explorer`. The status bar at the bottom indicates the date as 04-05-2022.

```
<@model OnlineAdz.Models.UserAd>
<div>
    <table border="1">
        <tr>
            <td>Email</td>
            <td>Ad Name</td>
        </tr>
        <tr>
            <td><@Html.DisplayFor(model => model.Email)></td>
            <td><@Html.DisplayFor(model => model.AdName)></td>
        </tr>
        <tr>
            <td>Ad Status</td>
            <td><@Html.DisplayFor(model => model.AdStatus)></td>
        </tr>
        <tr>
            <td>User Interest</td>
            <td><@Html.DisplayFor(model => model.UserInterest)></td>
        </tr>
    </table>
</div>
<div>
    <a href="#">Edit</a> |
    <a href="#">Index</a> |
    <a href="#">Back to List</a>
</div>
```

Edit:

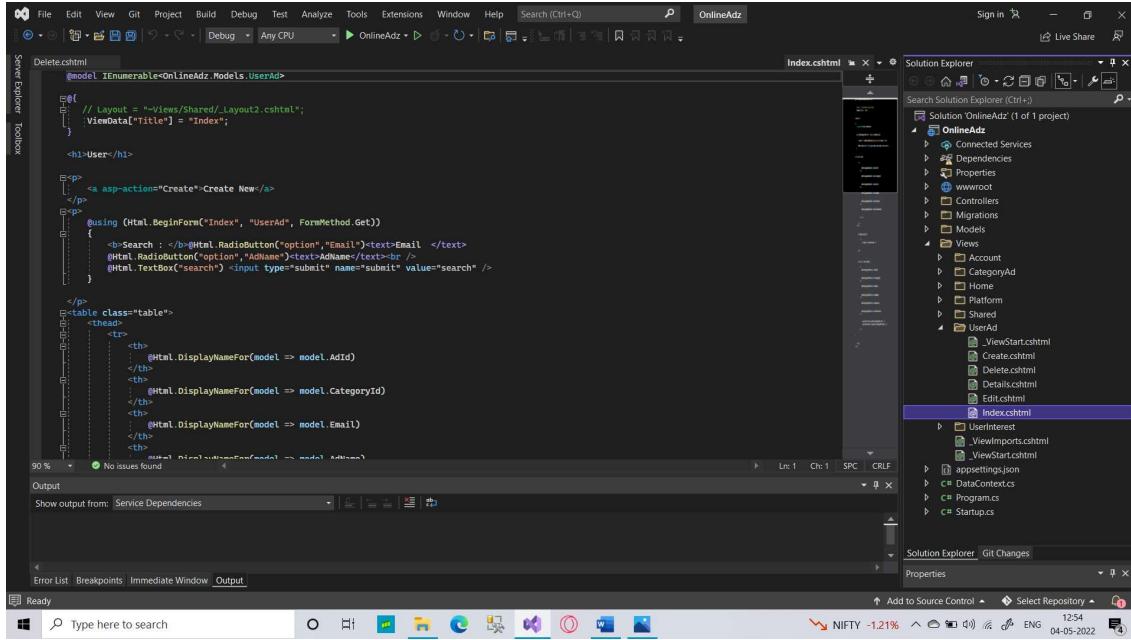
The screenshot shows the Visual Studio IDE interface with the following details:

- Title Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), OnlineAdz.
- Solution Explorer:** Shows the project structure for "OnlineAdz" with files like Delete.cshtml, Edit.cshtml, Index.cshtml, etc., under the "Views/UserAd" folder.
- Code Editor:** Displays the "Edit.cshtml" file content. The code is a form-based view for editing user information. It includes sections for "Category", "Email", "AdName", and "AdStatus". Validation messages are placed near the respective input fields.
- Status Bar:** Shows "90 %", "No issues found", "Ln: 1 Ch: 1 SPC CRLF", and a timestamp of "12:54 04-05-2022".

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

- Title Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), OnlineAdz.
- Solution Explorer:** Shows the project structure for "OnlineAdz" with files like Delete.cshtml, Edit.cshtml, Index.cshtml, etc., under the "Views/UserAd" folder.
- Code Editor:** Displays the "Edit.cshtml" file content. This version of the code uses a more simplified approach with a single large form group containing all fields. It includes validation messages and a submit button.
- Status Bar:** Shows "90 %", "No issues found", "Ln: 1 Ch: 1 SPC CRLF", and a timestamp of "12:54 04-05-2022".

## Index:



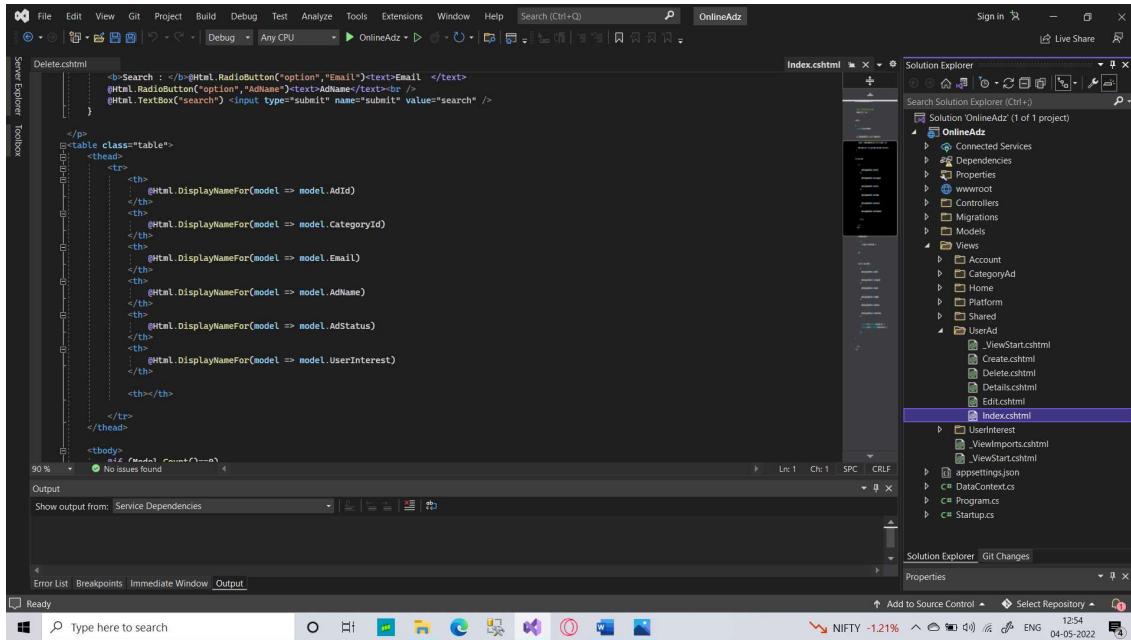
```
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) OnlineAdz
Delete.cshtml
Index.cshtml
Server Explorer
Toolbox
Solution Explorer
Search Solution Explorer (Ctrl+X)
Solution OnlineAdz (1 of 1 project)
OnlineAdz
Connected Services
Dependencies
Properties
wwwroot
Controllers
Migrations
Models
Views
Account
CategoryAd
Home
Platform
Shared
UserAd
ViewStart.cshtml
Create.cshtml
Delete.cshtml
Details.cshtml
Edit.cshtml
Index.cshtml
UserInterest
ViewImports.cshtml
ViewStart.cshtml
appsettings.json
DataContext.cs
Program.cs
Startup.cs
Solution Explorer_Git Changes
Properties
Add to Source Control Select Repository
Output
Show output from: Service Dependencies
Error List Breakpoints Immediate Window Output
Ready
Type here to search
NIFTY -1.21% 12:54 ENG 04-05-2022
```

```
<model as="IEnumerable<OnlineAdz.Models.UserAd>>
@{
    // Layout = "~/Views/Shared/_Layout2.cshtml";
    ViewData["Title"] = "Index";
}

<h1>User</h1>

<p><a href="#" asp-action="Create">Create New</a></p>
<form asp-action="Index" asp-method="Get">
    <div>
        <input type="text" name="search" value="search" />
    </div>
</form>

<table class="table">
    <thead>
        <tr>
            <th>@Html.DisplayNameFor(model => model.AdId)</th>
            <th>@Html.DisplayNameFor(model => model.CategoryId)</th>
            <th>@Html.DisplayNameFor(model => model.Email)</th>
            <th>@Html.DisplayNameFor(model => model.AdName)</th>
            <th>@Html.DisplayNameFor(model => model.AdStatus)</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>@Html.RadioButton("option", "Email")<text>Email</text></td>
            <td>@Html.RadioButton("option", "AdName")<text>AdName</text><br /><input type="text" name="search" value="search" /></td>
            <td>@Html.TextBox("search")<input type="submit" name="submit" value="search" /></td>
            <td></td>
            <td></td>
        </tr>
    </tbody>
</table>
```



```
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) OnlineAdz
Delete.cshtml
Index.cshtml
Server Explorer
Toolbox
Solution Explorer
Search Solution Explorer (Ctrl+X)
Solution OnlineAdz (1 of 1 project)
OnlineAdz
Connected Services
Dependencies
Properties
wwwroot
Controllers
Migrations
Models
Views
Account
CategoryAd
Home
Platform
Shared
UserAd
ViewStart.cshtml
Create.cshtml
Delete.cshtml
Details.cshtml
Edit.cshtml
Index.cshtml
UserInterest
ViewImports.cshtml
ViewStart.cshtml
appsettings.json
DataContext.cs
Program.cs
Startup.cs
Solution Explorer_Git Changes
Properties
Add to Source Control Select Repository
Output
Show output from: Service Dependencies
Error List Breakpoints Immediate Window Output
Ready
Type here to search
NIFTY -1.21% 12:54 ENG 04-05-2022
```

```
<model as="IEnumerable<OnlineAdz.Models.UserAd>>
@{
    // Layout = "~/Views/Shared/_Layout2.cshtml";
    ViewData["Title"] = "Index";
}

<p><a href="#" asp-action="Create">Create New</a></p>
<form asp-action="Index" asp-method="Get">
    <div>
        <input type="text" name="search" value="search" />
    </div>
</form>

<table class="table">
    <thead>
        <tr>
            <th>@Html.DisplayNameFor(model => model.AdId)</th>
            <th>@Html.DisplayNameFor(model => model.CategoryId)</th>
            <th>@Html.DisplayNameFor(model => model.Email)</th>
            <th>@Html.DisplayNameFor(model => model.AdName)</th>
            <th>@Html.DisplayNameFor(model => model.AdStatus)</th>
            <th>@Html.DisplayNameFor(model => model.UserInterest)</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>@Html.RadioButton("option", "Email")<text>Email</text></td>
            <td>@Html.RadioButton("option", "AdName")<text>AdName</text><br /><input type="text" name="search" value="search" /></td>
            <td>@Html.TextBox("search")<input type="submit" name="submit" value="search" /></td>
            <td></td>
            <td></td>
            <td></td>
        </tr>
    </tbody>
</table>
```

```

<tr>
    <td colspan="6" style="text-align: center;">No match found
</tr>
}
else
{
    foreach (var item in Model)
    {
        <tr>
            <td>@Html.DisplayFor(modelItem => item.AdId)</td>
            <td>@Html.DisplayFor(modelItem => item.CategoryId)</td>
            <td>@Html.DisplayFor(modelItem => item.Email)</td>
            <td>@Html.DisplayFor(modelItem => item.AdName)</td>
            <td>@Html.DisplayFor(modelItem => item.AdStatus)</td>
            <td>
                @Html.ActionLink("Edit", "Edit", new { id = item.AdId }) |
                @Html.ActionLink("Details", "Details", new { id = item.AdId }) |
                @Html.ActionLink("Delete", "Delete", new { id = item.AdId })
            </td>
        </tr>
    }
}

```

## Views: UserInterest:

### Index:

```

<model OnlineAdz.Models.UserInterest>
    <@ ViewData["Title"] = "Index"; >
<!DOCTYPE html>
<html>
    <head>
        <meta name="viewport" content="width=device-width"/>
    </head>
    <body>
        <using (Html.BeginForm("Index", "UserInterest", FormMethod.Post))>
            <table cellpadding="4px" cellspacing="30px">
                <tr>
                    <td><h5>Email : </h5></td>
                    <td>@Html.TextBoxFor(m => m.Email)</td>
                </tr>
                <tr>
                    <td><h5>Category Id : </h5></td>
                    <td>@Html.TextBoxFor(m => m.CategoryId)</td>
                </tr>
                <tr>
                    <td><h5>Ad Name : </h5></td>
                    <td>@Html.TextBoxFor(m => m.AdName)</td>
                </tr>
            </table>
        </using>
    </body>
</html>

```

```
<tr>
    <td><h5>Category Id:</h5></td>
    <td>
        @Html.TextBoxFor(m => m.CategoryId)
    </td>
</tr>
<tr>
    <td><h5>Ad Name:</h5></td>
    <td>
        @Html.TextBoxFor(m => m.AdName)
    </td>
</tr>
<tr>
    <td><h5>Ad Desc.:</h5></td>
    <td>
        @Html.TextBoxFor(m => m.AdDesc)
    </td>
</tr>
<tr>
    <td></td>
    <td><input type="submit" value="Submit" style="background-color:blue;color:white;border:none;" /></td>
</tr>
}
</table>
</body>
</html>
```

No issues found

Output

Error List Breakpoints Immediate Window Output

Solution Explorer

OnlineAdz

Views

- Index.cshtml
- \_ViewImports.cshtml
- \_ViewStart.cshtml

Properties

Startup.cs

Program.cs

DataContext.cs

appsettings.json

UserAd

CategoryAd

Home

Platform

Shared

Account

BHARTIARTL -1.75% 12:56 ENG 04-05-2022

## Views:ViewStart:

```
Layout = "_Layout";
```

No issues found

Output

Error List Breakpoints Immediate Window Output

Solution Explorer

OnlineAdz

Views

- \_ViewStart.cshtml
- Index.cshtml
- \_ViewImports.cshtml
- \_ViewStart.cshtml

Properties

Startup.cs

Program.cs

DataContext.cs

appsettings.json

UserAd

CategoryAd

Home

Platform

Shared

Account

BHARTIARTL -1.75% 12:56 ENG 04-05-2022

## Appsettings.json:

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

- Title Bar:** Shows "OnlineAdz" as the project name.
- Solution Explorer:** Displays the project structure for "OnlineAdz".
- Appsettings.json File:** Opened in the center pane, showing configuration settings for the application.
- Output Window:** Shows "No issues found".
- Taskbar:** Includes icons for File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and Search (Ctrl+Q).
- System Tray:** Shows battery status, network connection, and system date/time (12:58, 04-05-2022).

```
Schema: https://json.schemastore.org/appsettings.json
{
  "ConnectionStrings": {
    "DefaultConnection": "Data Source=.;Database=OnlineAdz; uid=sa;pwd=12345;"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Warning"
    }
  },
  "AllowedHosts": "*"
}
```

## DataContext:

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

- Title Bar:** Shows "OnlineAdz" as the project name.
- Solution Explorer:** Displays the project structure for "OnlineAdz".
- DataContext.cs File:** Opened in the center pane, showing the implementation of the DataContext class.
- Output Window:** Shows "No issues found".
- Taskbar:** Includes icons for File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and Search (Ctrl+Q).
- System Tray:** Shows battery status, network connection, and system date/time (12:58, 04-05-2022).

```
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore;

namespace OnlineAdz
{
  public class DataContext : IdentityDbContext<IdentityUser, IdentityRole, string, IdentityUserClaim<string>, IdentityUserRole<string>>
  {
    public DataContext(DbContextOptions<DataContext> options)
      : base(options)
    {
    }

    protected override void OnModelCreating(ModelBuilder builder)
    {
      base.OnModelCreating(builder);
      builder.Entity<IdentityUser>();
    }
  }
}
```

## Program:

The screenshot shows the Visual Studio IDE interface. The code editor displays the `Program.cs` file for an ASP.NET Core application named `OnlineAdz`. The file contains the following code:

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore;
using Microsoft.AspNetCore.Hosting;
```

```
namespace OnlineAdz
{
    public class Program
    {
        public static void Main(string[] args)
        {
            BuildWebHost(args).Run();
        }

        public static IWebHost BuildWebHost(string[] args) =>
            webHost.CreateDefaultBuilder(args)
                .UseStartup<Startup>()
                .Build();
    }
}
```

The Solution Explorer on the right shows the project structure for `Solution OnlineAdz`, which includes files like `Connected Services`, `Dependencies`, `Properties`, `wwwroot`, `Controllers`, `Migrations`, `Models`, `Views` (containing `Account`, `CategoryAd`, `Home`, `Platform`, `Shared`, `UserAd`, and `UserInterest`), `appsettings.json`, `Contexts.cs`, `Program.cs`, and `Startup.cs`.

## StartUp:

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

- Title Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), OnlineAdz.
- Solution Explorer:** Shows the solution "OnlineAdz" with one project. The project structure includes: Connected Services, Dependencies, Properties, wwwroot, Controllers, Migrations, Models, Views, Account, CategoryAd, Home, Platform, Shared, UserAd, and UserDetail. It also lists \_ViewStart.cshtml, Index.cshtml, \_ViewImports.cshtml, and \_ViewStart2.cshtml under Views.
- Code Editor:** The current file is Startup.cs, which contains the code for the ASP.NET Core application. The code includes imports for Microsoft.AspNetCore.Builder, Microsoft.AspNetCore.Hosting, Microsoft.AspNetCore.Http, Microsoft.AspNetCore.HttpOverrides, Microsoft.AspNetCore.Identity, Microsoft.EntityFrameworkCore, Microsoft.Extensions.Configuration, Microsoft.Extensions.DependencyInjection, Microsoft.Extensions.Hosting, and OnlineAdz.Models. The class Startup is defined with a constructor taking IConfiguration and IWebHostEnvironment. Inside the constructor, Configuration is assigned to configuration, environment is assigned to env, and a ConfigurationBuilder is created. The builder is used to set the base path, add JSON files for appsettings.json and appsettings.{env.EnvironmentName}.json, and add environment variables. Finally, the configuration is built. The code editor shows syntax highlighting and some refactoring tools.
- Status Bar:** Shows the status "99%" and "No issues found".
- Output Tab:** Shows the output from "Service Dependencies".
- Search Bar:** Type here to search.

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

- Title Bar:** OnlineAdz
- Toolbars:** Standard (File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help), Search (Ctrl+Q), Live Share.
- Code Editor:** Startup.cs (Configuration configuration, IWebHostEnvironment env) - This is the main entry point for the application. It includes configuration setup, service registration, and pipeline configuration.
- Solution Explorer:** Shows the project structure for "Solution OnlineAdz" (1 of 1 project). It includes files like appsettings.json, Program.cs, and various controller and view files.
- Properties:** Shows the properties for the selected file.
- Output:** Shows no issues found.
- Taskbar:** Includes icons for File Explorer, Task View, Start, Taskbar settings, and a search bar.
- System Tray:** Shows the date and time (04-05-2022), battery level (BHARTIARTL -1.75%), and other system status indicators.

```
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace OnlineAdz
{
    public class Startup
    {
        public IConfiguration Configuration { get; }
        public IWebHostEnvironment Environment { get; }

        public Startup(IConfiguration configuration, IWebHostEnvironment env)
        {
            Configuration = configuration;
            Environment = env;
            var builder = new ConfigurationBuilder()
                .SetBasePath(env.ContentRootPath)
                .AddJsonFile("appsettings.json", optional: true, reloadOnChange: true)
                .AddJsonFile($"appsettings.{env.EnvironmentName}.json", optional: true)
                .AddEnvironmentVariables();
            Configuration = builder.Build();
        }

        public void ConfigureServices(IServiceCollection services)
        {
            services.AddControllersWithViews();
            services.AddDbContext<DataContext>(
                options => options.UseSqlServer("name=ConnectionStrings:DefaultConnection"));
            //services.AddIdentity<ApplicationUser, Role>().AddEntityFrameworkStores<DbContext>();
            services.AddIdentity<User, IdentityRole>()
                .AddEntityFrameworkStores<DbContext>()
                .AddDefaultTokenProviders();

            services.Configure<CookiePolicyOptions>(options =>
            {
                // This lambda determines whether user consent for non-essential cookies is needed for a given request.
                options.CheckConsentNeeded = context => true;
                options.MinimumSameSitePolicy = SameSiteMode.None;
            });
            services.AddSession(options =>
            {
                options.Cookie.SameSite = SameSiteMode.None;
                options.Cookie.SecurePolicy = CookieSecurePolicy.Always;
                options.Cookie.IsEssential = true;
            });
        }

        public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
        {
            if (env.IsDevelopment())
            {
                app.UseDeveloperExceptionPage();
            }
            else
            {
                app.UseExceptionHandler("/Error");
                app.UseHsts();
            }

            app.UseHttpsRedirection();
            app.UseStaticFiles();
            app.UseRouting();
            app.UseAuthorization();
            app.UseEndpoints(endpoints =>
            {
                endpoints.MapControllerRoute(
                    name: "default",
                    pattern: "{controller}/{action}/{id?}");
                endpoints.MapControllerRoute(
                    name: "areas",
                    pattern: "{area}/{controller}/{action}/{id?}");
            });
        }
    }
}
```

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

- Title Bar:** OnlineAdz
- Toolbars:** Standard (File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), Live Share).
- Code Editor:** Startup.cs (Configuration configuration, IWebHostEnvironment env) - This is the main entry point for the application. It includes configuration setup, service registration, and pipeline configuration.
- Solution Explorer:** Shows the project structure for "Solution OnlineAdz" (1 of 1 project). It includes files like appsettings.json, Program.cs, and various controller and view files.
- Properties:** Shows the properties for the selected file.
- Output:** Shows no issues found.
- Taskbar:** Includes icons for File Explorer, Task View, Start, Taskbar settings, and a search bar.
- System Tray:** Shows the date and time (04-05-2022), battery level (BHARTIARTL -1.75%), and other system status indicators.

```
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace OnlineAdz
{
    public class Startup
    {
        public IConfiguration Configuration { get; }
        public IWebHostEnvironment Environment { get; }

        public Startup(IConfiguration configuration, IWebHostEnvironment env)
        {
            Configuration = configuration;
            Environment = env;
            var builder = new ConfigurationBuilder()
                .SetBasePath(env.ContentRootPath)
                .AddJsonFile("appsettings.json", optional: true, reloadOnChange: true)
                .AddJsonFile($"appsettings.{env.EnvironmentName}.json", optional: true)
                .AddEnvironmentVariables();
            Configuration = builder.Build();
        }

        public void ConfigureServices(IServiceCollection services)
        {
            services.AddControllersWithViews();
            services.AddDbContext<DataContext>(
                options => options.UseSqlServer("name=ConnectionStrings:DefaultConnection"));
            //services.AddIdentity<ApplicationUser, Role>().AddEntityFrameworkStores<DbContext>();
            services.AddIdentity<User, IdentityRole>()
                .AddEntityFrameworkStores<DbContext>()
                .AddDefaultTokenProviders();

            services.Configure<CookiePolicyOptions>(options =>
            {
                // This lambda determines whether user consent for non-essential cookies is needed for a given request.
                options.CheckConsentNeeded = context => true;
                options.MinimumSameSitePolicy = SameSiteMode.None;
            });
            services.AddSession(options =>
            {
                options.Cookie.SameSite = SameSiteMode.None;
                options.Cookie.SecurePolicy = CookieSecurePolicy.Always;
                options.Cookie.IsEssential = true;
            });
        }

        public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
        {
            if (env.IsDevelopment())
            {
                app.UseDeveloperExceptionPage();
            }
            else
            {
                app.UseExceptionHandler("/Error");
                app.UseHsts();
            }

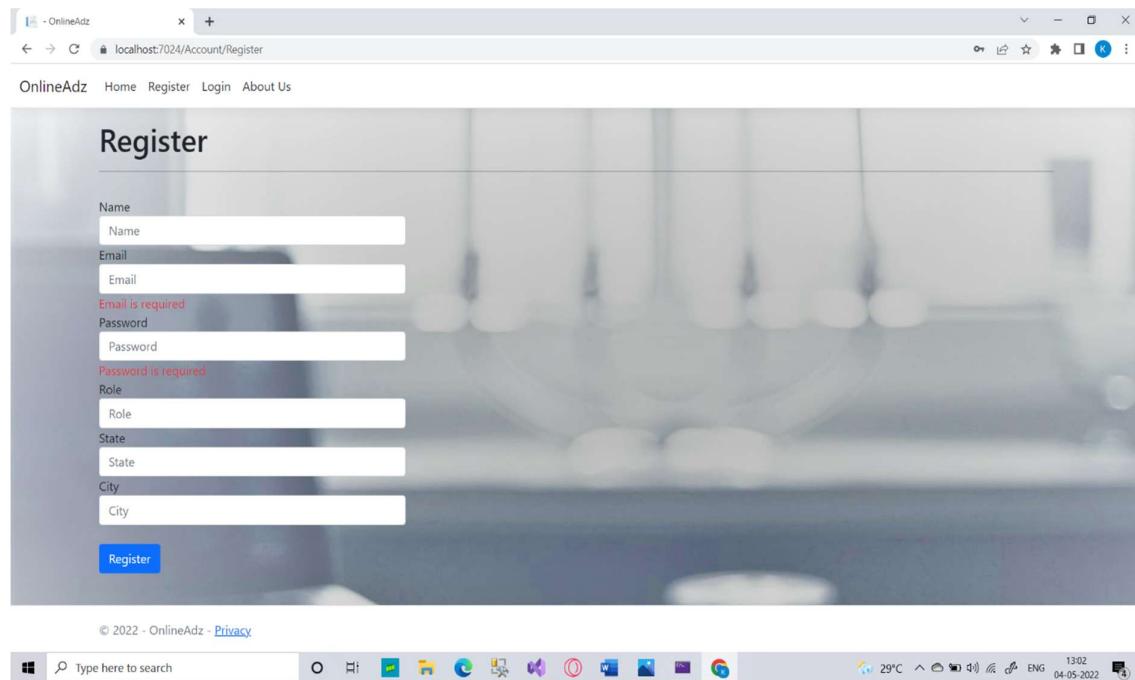
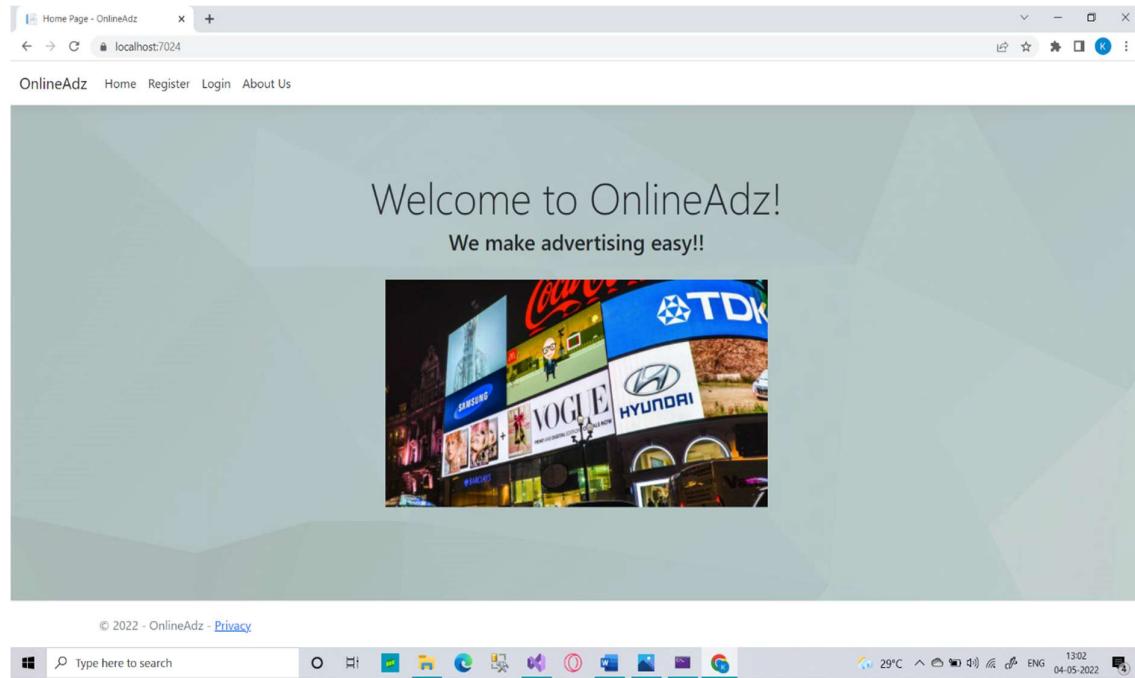
            app.UseHttpsRedirection();
            app.UseStaticFiles();
            app.UseRouting();
            app.UseAuthorization();
            app.UseEndpoints(endpoints =>
            {
                endpoints.MapControllerRoute(
                    name: "default",
                    pattern: "{controller}/{action}/{id?}");
                endpoints.MapControllerRoute(
                    name: "areas",
                    pattern: "{area}/{controller}/{action}/{id?}");
            });
        }
    }
}
```

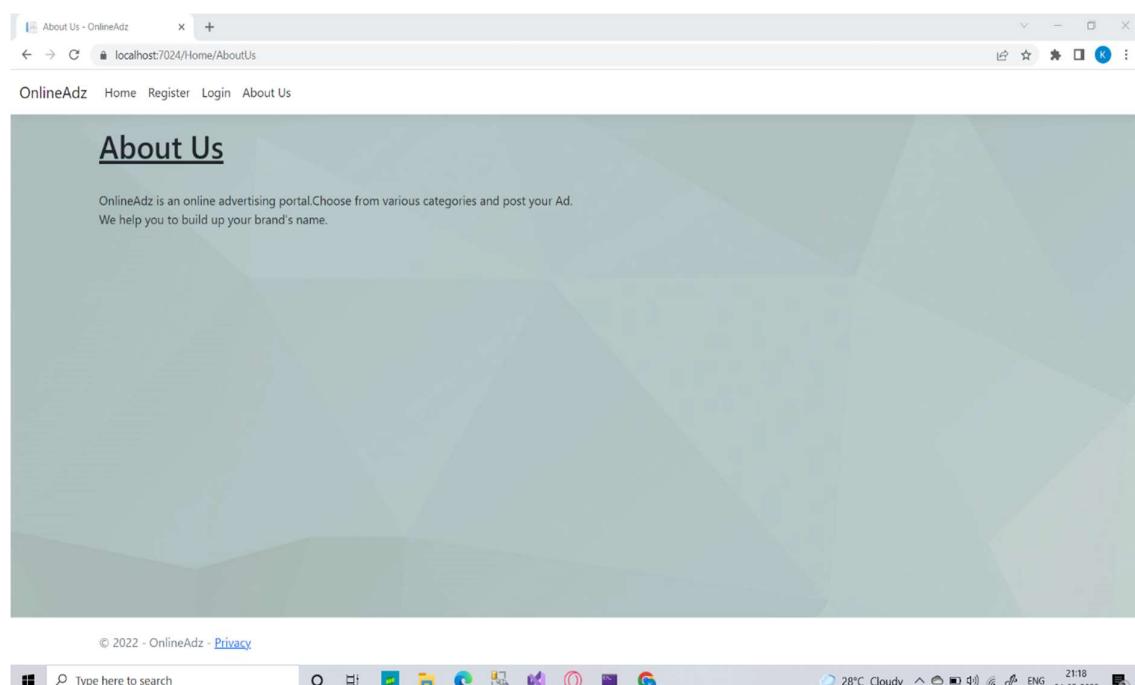
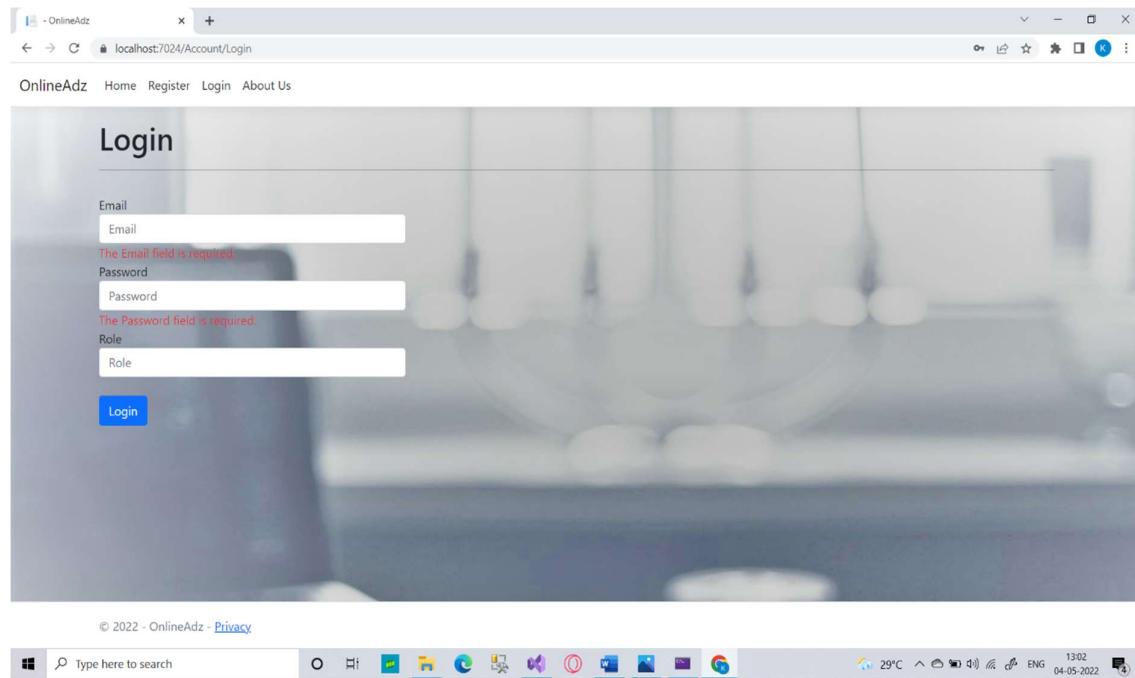
The screenshot shows the Microsoft Visual Studio IDE interface. The main area displays the `Startup.cs` file for a .NET Core application named `OnlineAdz`. The code implements the `IWebHostBuilder` interface to configure the ASP.NET Core application. It includes logic for developer exception pages, static files, routing, and endpoints. The `Configure` method sets up the `app` object with various middleware components like `UseDeveloperExceptionPage`, `UseStaticFiles`, `UseRouting`, `UseCors`, `UseAuthentication`, `UseAuthorization`, and `UseEndpoints`. The `UseEndpoints` call maps a controller route named "default" to the `Home` controller's `Index` action. The `Solution Explorer` window on the right lists the project structure, including `Startup.cs` and other files like `Program.cs` and `appsettings.json`. The bottom status bar shows the user's name as BHARTIARTL, battery level at 1.75%, and the date/time as 04-05-2022.

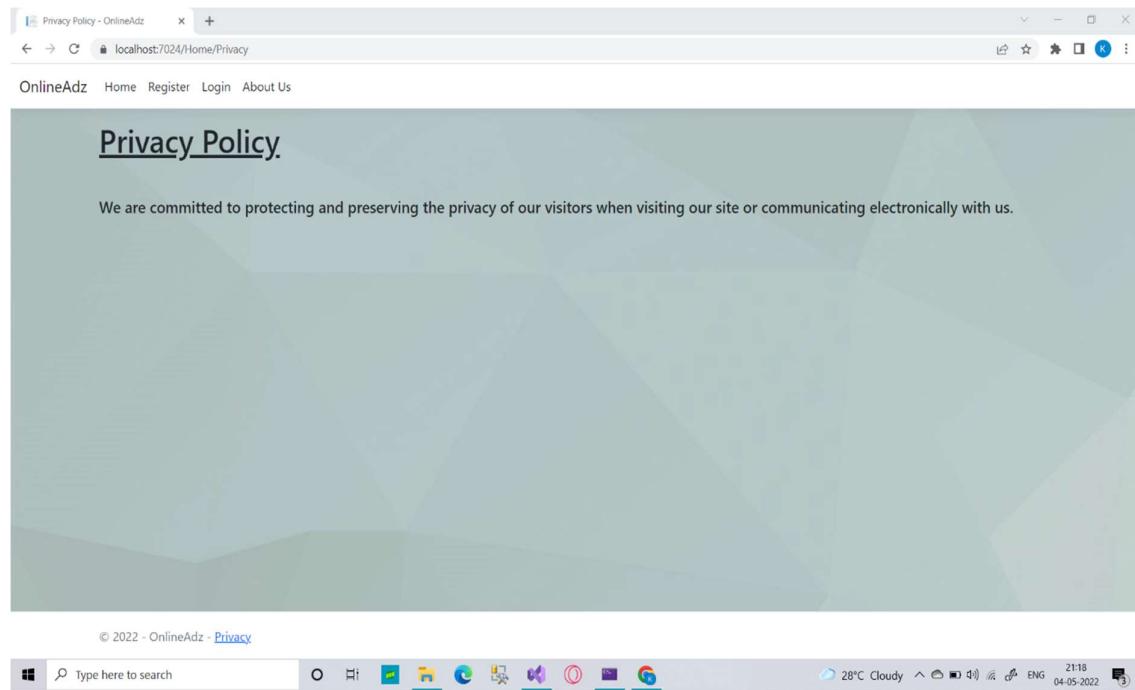
```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseExceptionHandler("/Home/Error");
        // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
        app.UseHsts();
    }
    app.UseStaticFiles();
    app.UseRouting();
    app.UseCors();
    app.UseAuthentication();
    app.UseAuthorization();
    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllerRoute(
            name: "default",
            pattern: "{controller=Home}/{action=Index}/{id?}");
    });
}
```

## Outputs:

For both Admin and User:



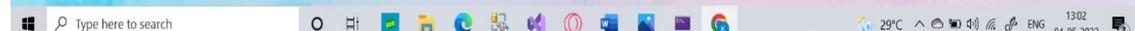
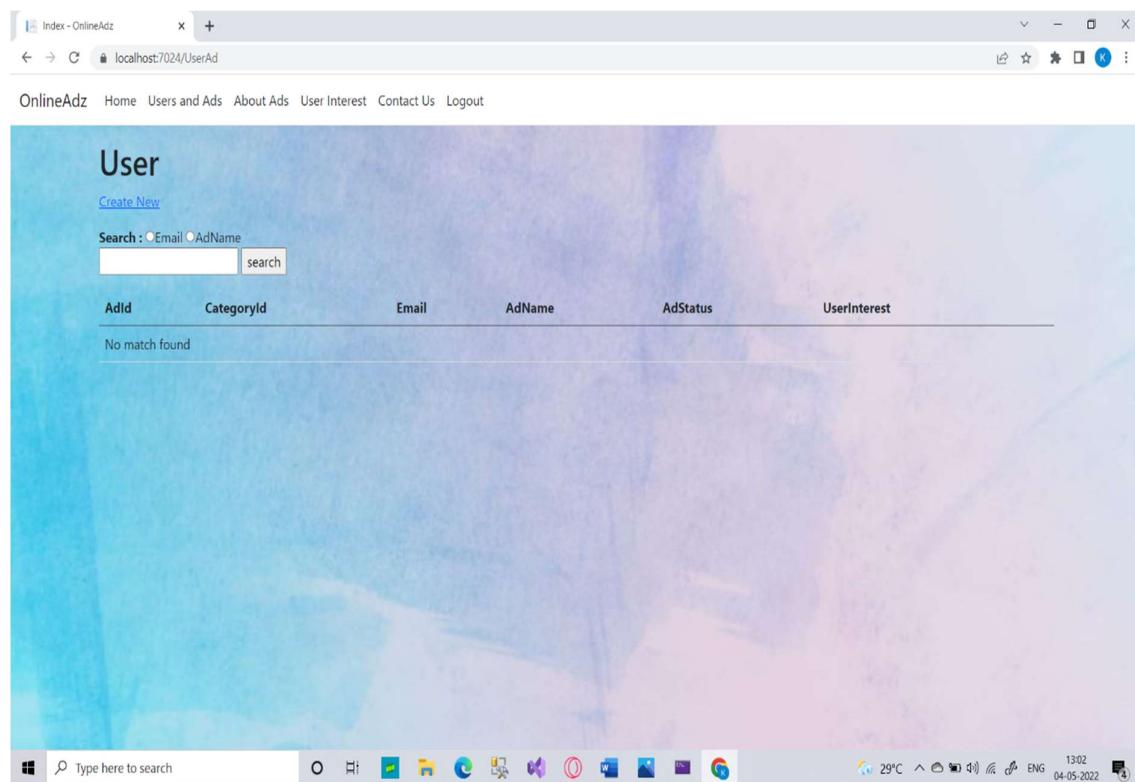




© 2022 - OnlineAdz - [Privacy](#)



## For User:



Index - OnlineAdz

localhost:7024/UserAd?option=Email&search=brandon%40gmail.com&submit=search

OnlineAdz Home Users and Ads About Ads User Interest Contact Us Logout

## User

Create New

Search : Email AdName  
brandon@gmail.com search

AdId	CategoryId	Email	AdName	AdStatus	UserInterest	
4	SM	brandon@gmail.com	Jasper Market	Active	Food and Beverages	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

Type here to search 28°C 2135 04-05-2022

Create - OnlineAdz

localhost:7024/UserAd/Create

OnlineAdz Home Register Login About Us

## Create

### Post Ad

CategoryId

Email

AdName

AdStatus

UserInterest

[Create](#) [Back to List](#)

© 2022 - OnlineAdz - [Privacy](#)

Type here to search 28°C 2137 04-05-2022

Edit - OnlineAdz

localhost:7024/UserAd/Edit/4

OnlineAdz Home Register Login About Us

## Edit

### Update Details

CategoryId	SM
Email	brandon@gmail.com
AdName	Jasper Market
AdStatus	Active
UserInterest	Food and Beverages

[Save](#) [Back to List](#)

© 2022 - OnlineAdz - [Privacy](#)



Details - OnlineAdz

localhost:7024/UserAd/Details/4

OnlineAdz Home Register Login About Us

## Details

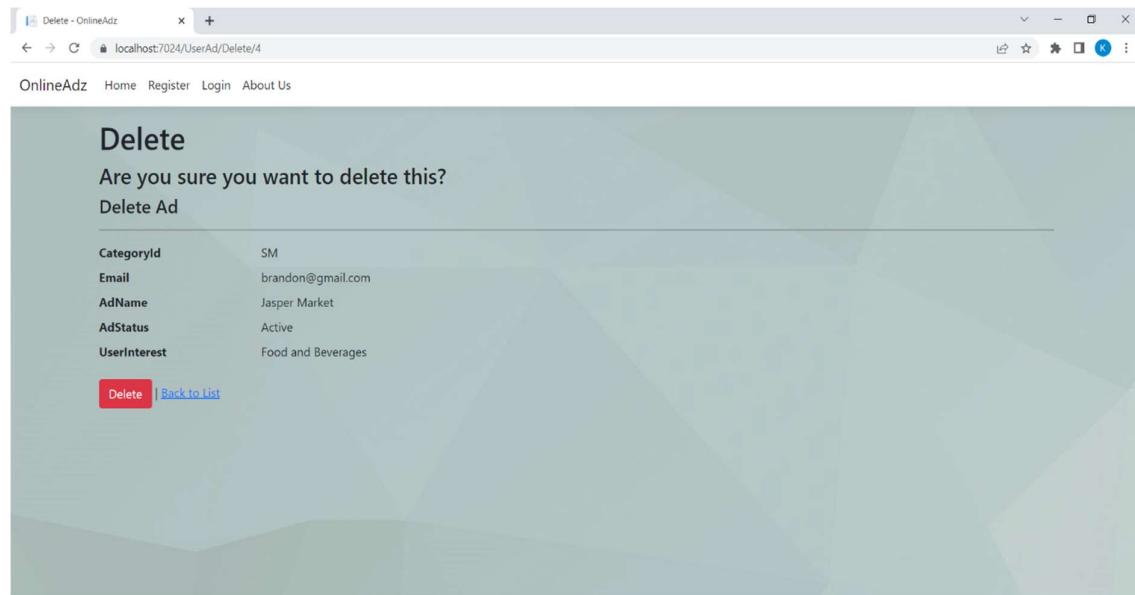
### Ad Details

CategoryId	SM
Email	brandon@gmail.com
AdName	Jasper Market
AdStatus	Active
UserInterest	Food and Beverages

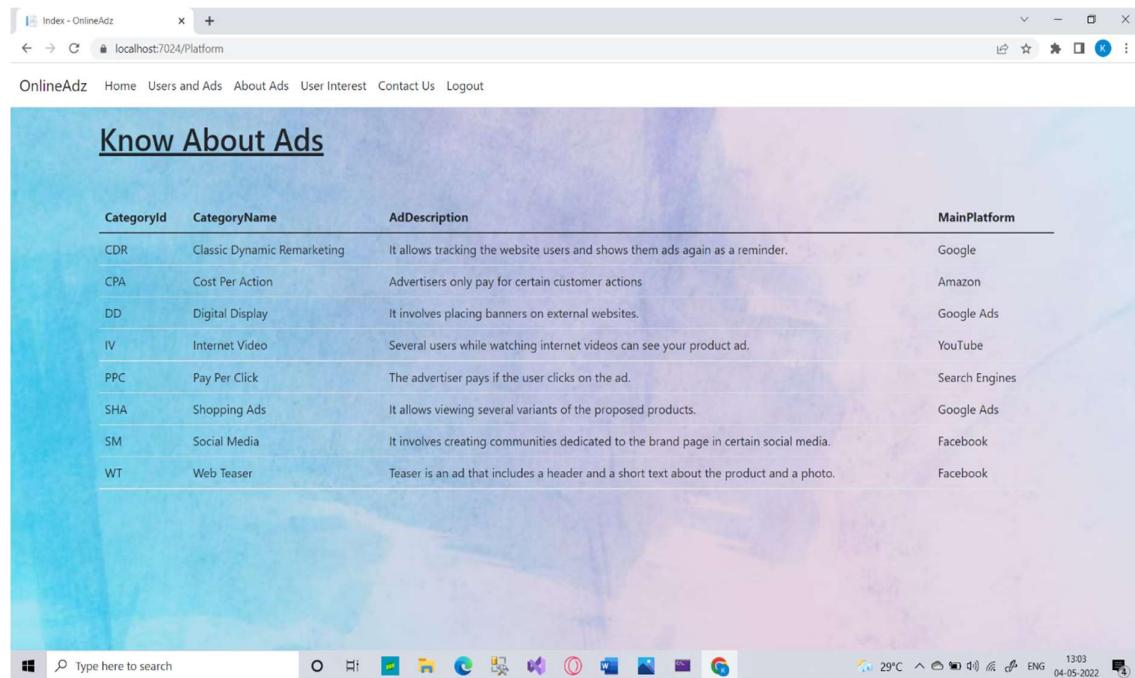
[Edit](#) | [Back to List](#)

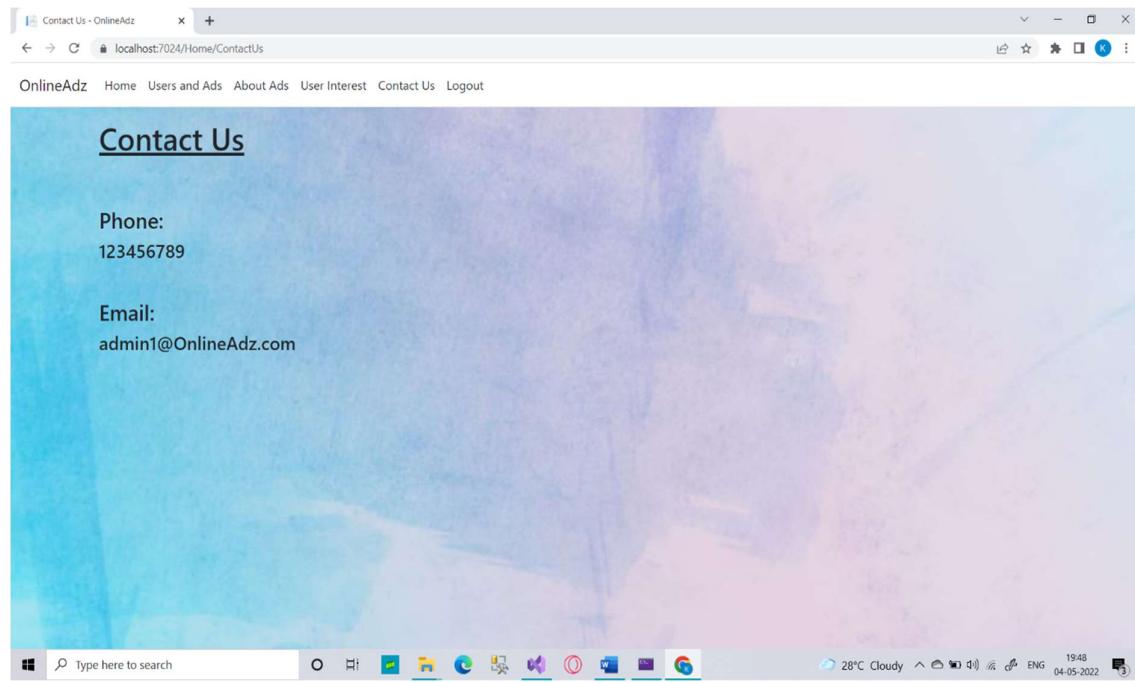
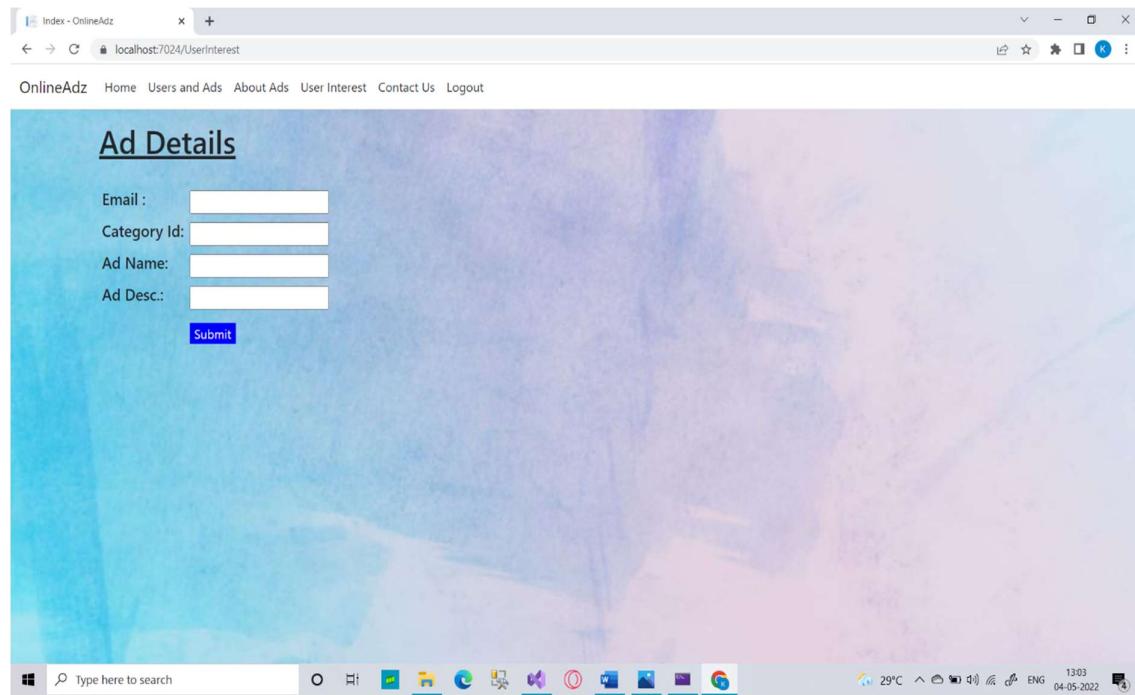
© 2022 - OnlineAdz - [Privacy](#)





© 2022 - OnlineAdz - [Privacy](#)





For Admin:

Index - OnlineAdz

localhost:7024/CategoryAd/Index

OnlineAdz Home Ad Category Users and Ads

## Types Of Ads

Create New

CategoryId	CategoryName	
CDR	Classic Dynamic Remarketing	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
CPA	Cost Per Action	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
DD	Digital Display	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
IV	Internet Video	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
PPC	Pay Per Click	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
SHA	Shopping Ads	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
SM	Social Media	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
WT	Web Teaser	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

Type here to search

29°C ENG 04-05-2022 13:04

Create - OnlineAdz

localhost:7024/CategoryAd/Create

OnlineAdz Home Register Login About Us

## Add new category

Ad Categories

CategoryId

CategoryName

**Create**

[Back to List](#)

© 2022 - OnlineAdz - [Privacy](#)

Type here to search

28°C ENG 04-05-2022 21:37

