

National University Manila
College of Computing and Information Technologies

**COURSE
PROJECT**

In Partial fulfillment in
BSIT-MWA
BS. INFORMATION TECHNOLOGY
WITH SPECIALIZATION IN MOBILE
AND WEB APPLICATION

Submitted by
Avila, Riv
Jardeleza, Johnrod
Mondonedo, Christian
Sabater, Gabriel

Rogel M. Labanan
Instructor

November 2023

Table of Contents

Project Overview..... 3

 Project Description..... 3

 Organization’s Profile..... 4

 Project Architecture (Network Topology)..... 5

Project Overview

Project Description

Our project is a NU Blogsite for students of MWA and MAA. Both students and faculty can view the blogsite and read the blogs. There is a designated webpage that only students of MWA and MAA can see and access, respectively. Students can post comments and edit their own comments while the faculty members have the accessibility to delete the students' comments. For the forums, students can create their own forum thread and discuss any topic that they like and it will be posted on their respective group and can only be viewed within the members of the group.

The project is developed using C# and ASP.NET in Visual Studio. The project is also developed within virtual machine environments, using tools such as OPNSense, Ubuntu, Windows Server, and Windows Client.

Organization's Profile

An active directory group within CCIT includes an IT group which is divided into two groups MWA & MAA which is also divided into two groups for Faculty and Students.

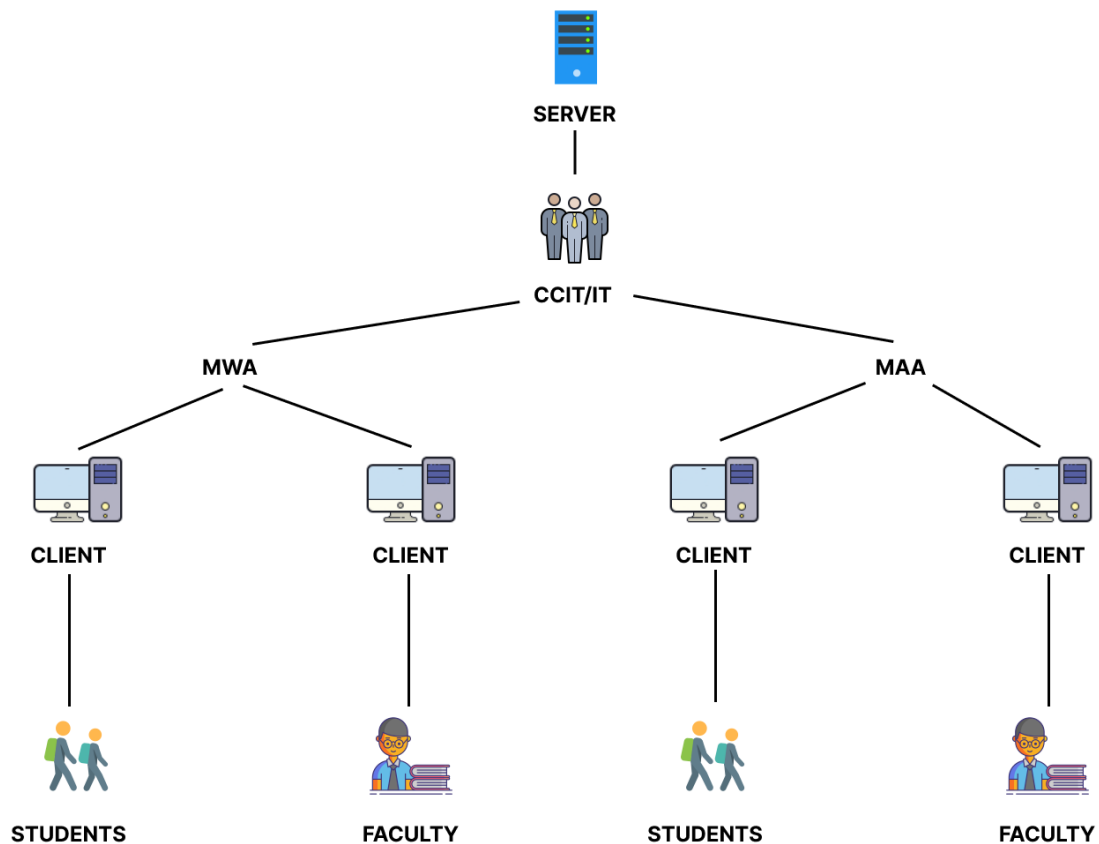
MWA

- Faculty
 - Under this group there are two users:
 - The faculty has special access to an MWA dedicated Webpage
 - Can delete students' comments
- Students
 - Under this group there are multiple users:
 - The students has special access to an MWA dedicated webpage
 - Can post comments on the forum page

MAA

- Faculty
 - Under this group there are two users:
 - The faculty has special access to an MAA dedicated Webpage
 - Can delete students' comments
- Students
 - Under this group there are multiple users:
 - The students has special access to an MAA dedicated webpage
 - Can post comments on the forum page

Project Architecture (Network Topology)



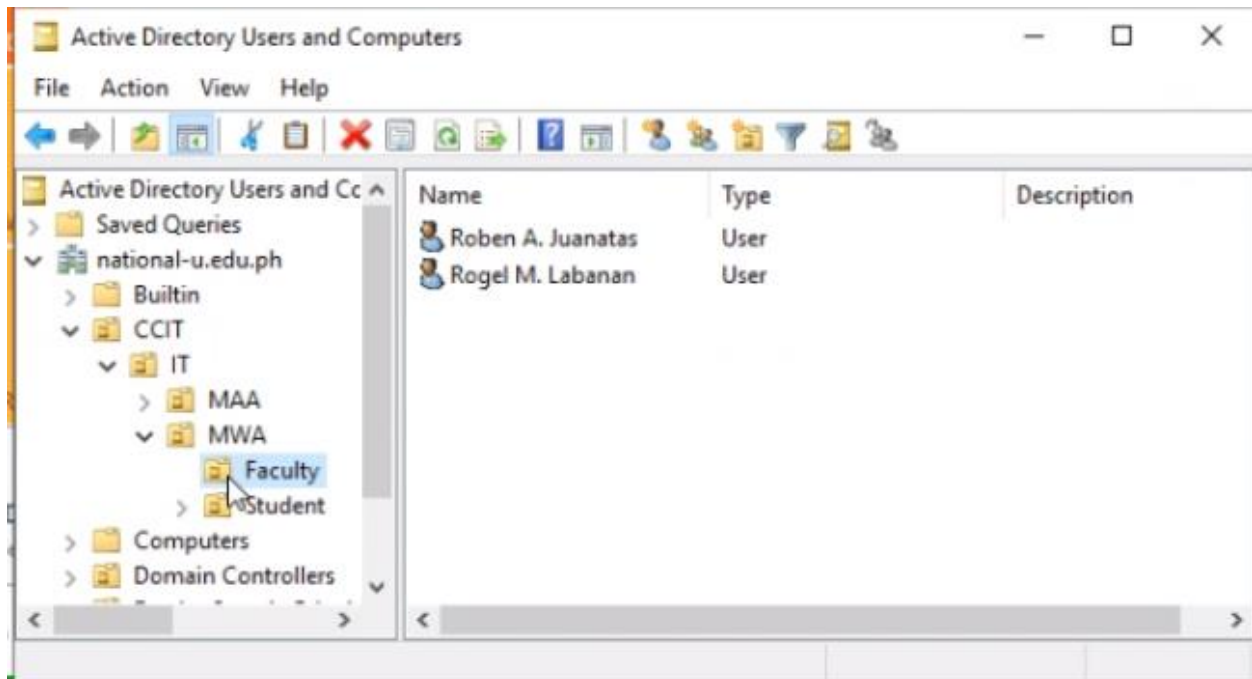
The project architecture of NU Blogsite. The website is locally hosted on the server. The server can be seen and accessed by the users within the group of MWA and MAA. Each user can login their accounts to every computer that is connected to the national-u.edu.ph domain in which users can also access and open the blogsite through the portal link (portal.national-u.edu.ph).

Implementation Documentation

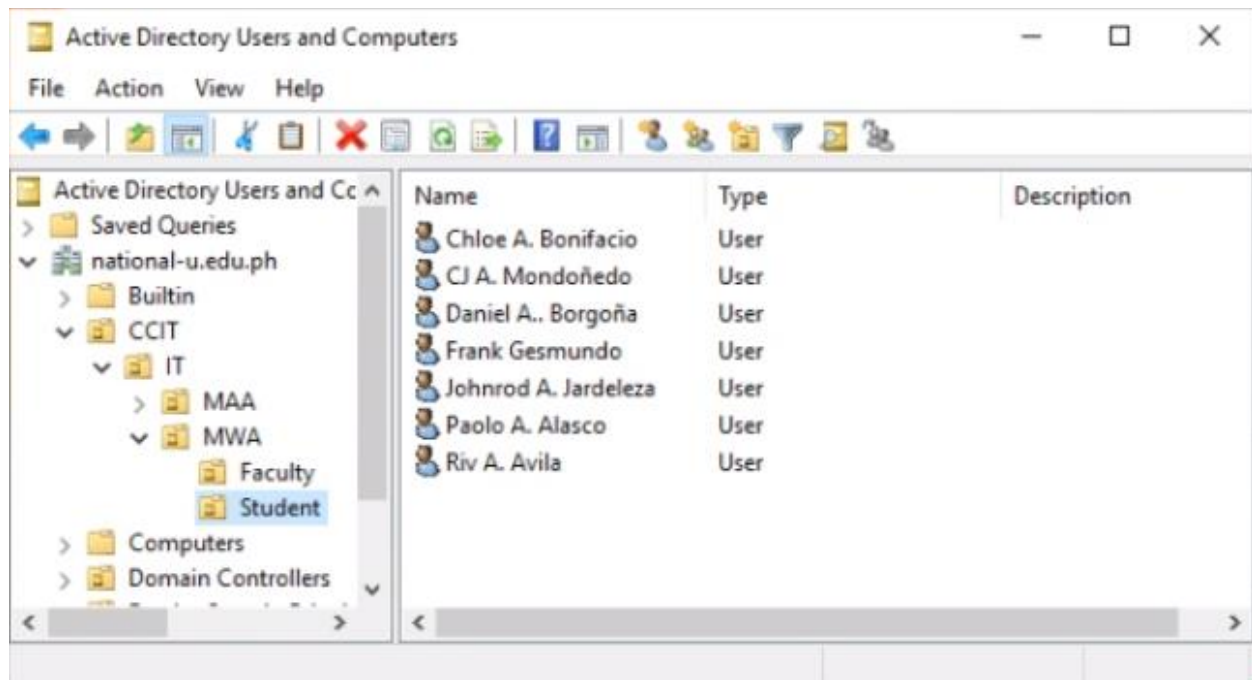
The blogsite was developed using C# and ASP.NET on Visual Studio. There are two main groups, which are MWA & MAA. Each group is divided into two groups between Faculty and Student.

MWA

Faculty

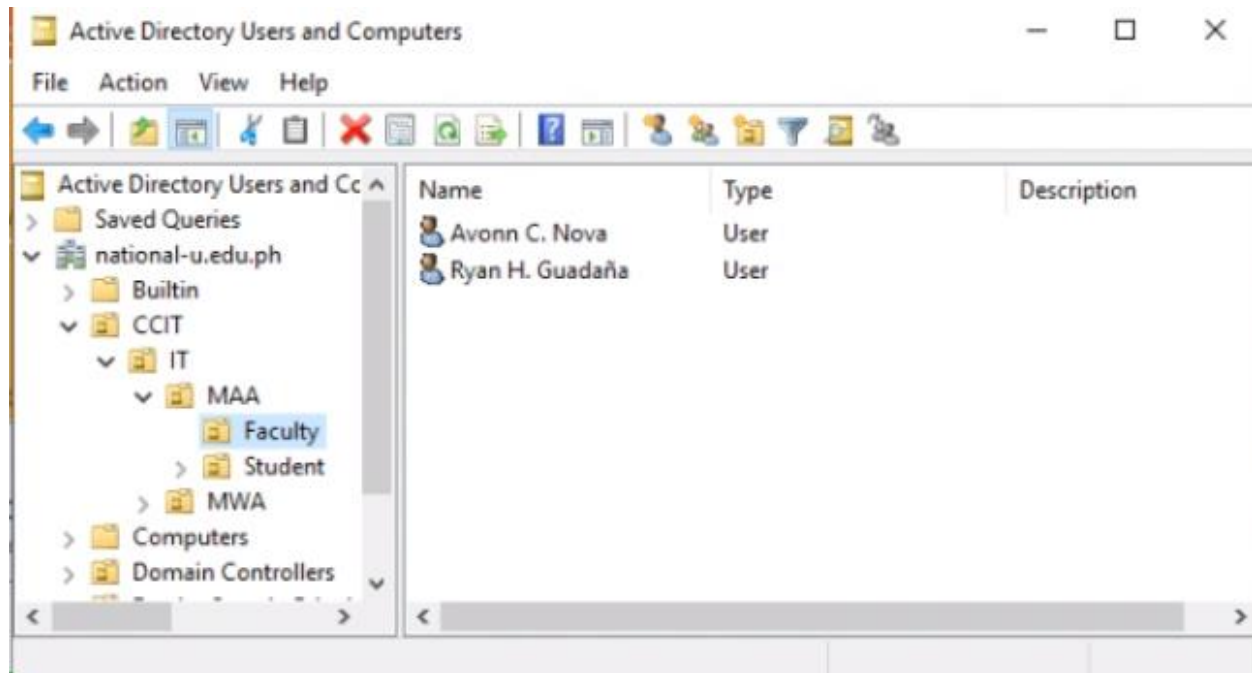


Student

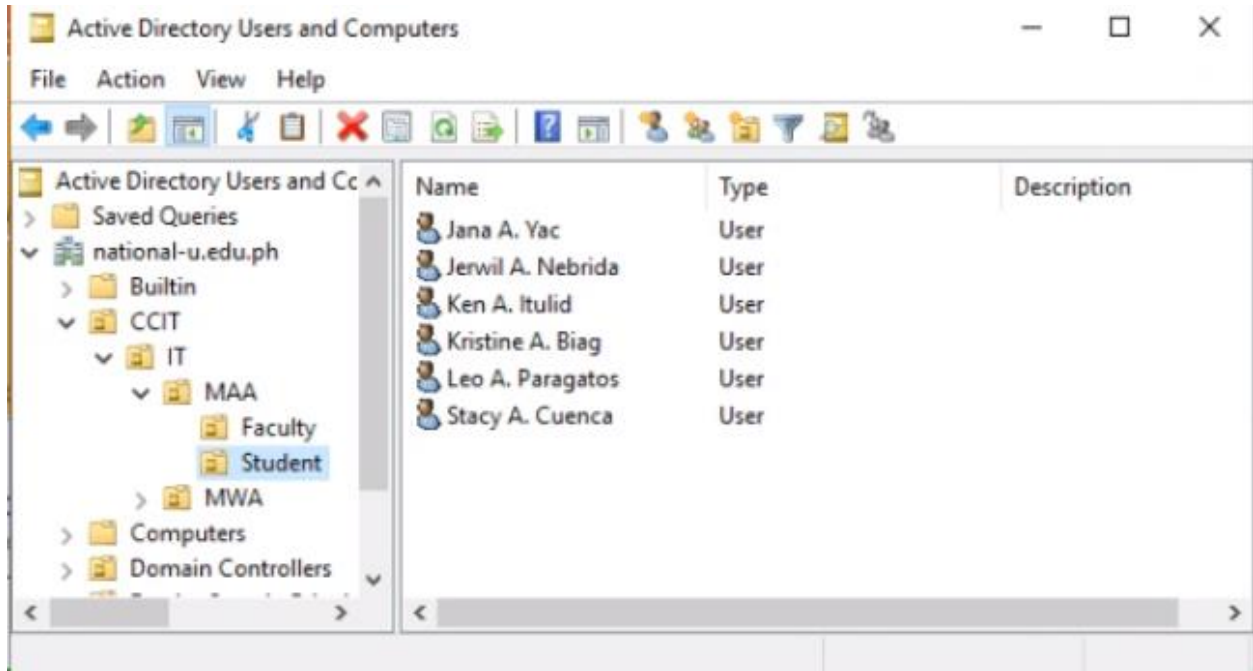


MAA

Faculty



Student



The implementation for the authorization for webpage access is shown below.


```
Index.cshtml | CommentsController.cs | _Layout.cshtml
BlogApp | BlogApp.Controllers.CommentsController | Edit(int id, Comments comments)

using Microsoft.AspNetCore.Authorization;

namespace BlogApp.Controllers
{
    [Authorize]
    1 reference
    public class CommentsController : Controller
    {
        private readonly BlogAppContext _context;

        0 references
        public CommentsController(BlogAppContext context)
        {
            _context = context;
        }

        // GET: Comments
        3 references
        public async Task<IActionResult> Index()
        {
            return _context.Comments != null ?
                View(await _context.Comments.ToListAsync()) :
                Problem("Entity set 'BlogAppContext.Comments' is null.");
        }
    }
}
```

```
Index.cshtml | CommentsController.cs | _Layout.cshtml
BlogApp | BlogApp.Controllers.CommentsController | Edit(int id, Comments comments)

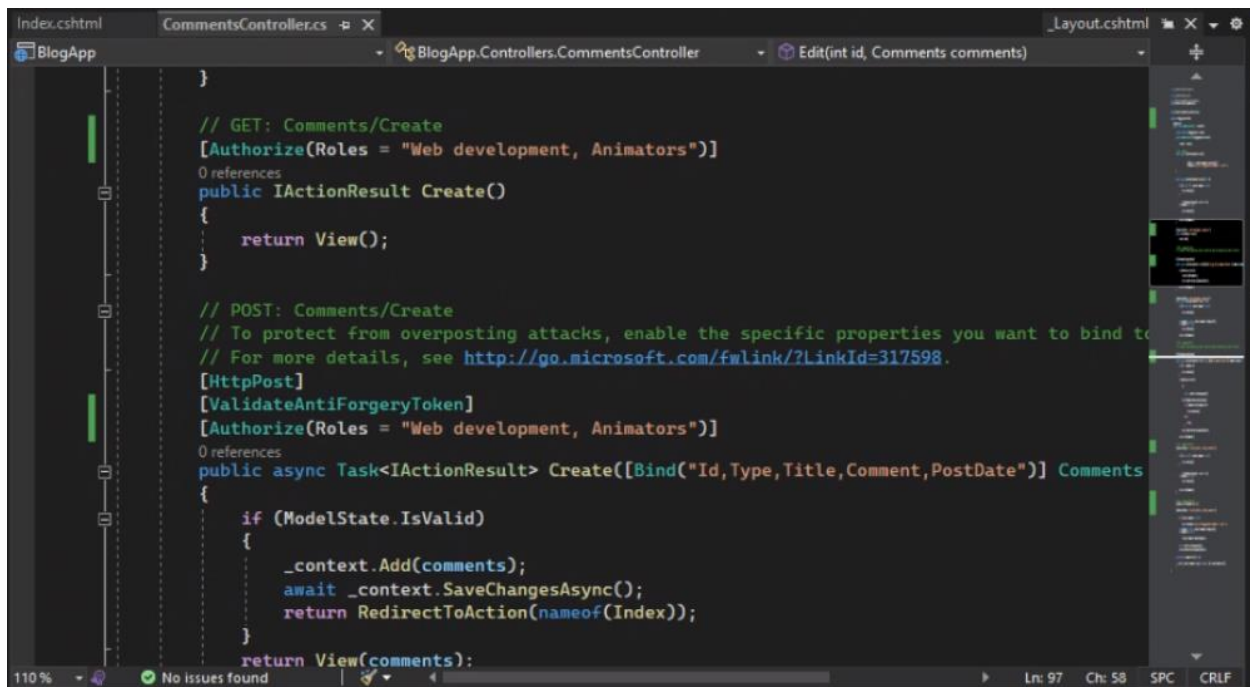
// GET: Comments/Details/5
0 references
public async Task<IActionResult> Details(int? id)
{
    if (id == null || _context.Comments == null)
    {
        return NotFound();
    }

    var comments = await _context.Comments
        .FirstOrDefaultAsync(m => m.Id == id);
    if (comments == null)
    {
        return NotFound();
    }

    return View(comments);
}

// GET: Comments/Create
[Authorize(Roles = "Web development, Animators")]
0 references
public IActionResult Create()
{
}
```

110 % | No issues found | Ln: 97 | Ch: 58 | SPC | CRLF



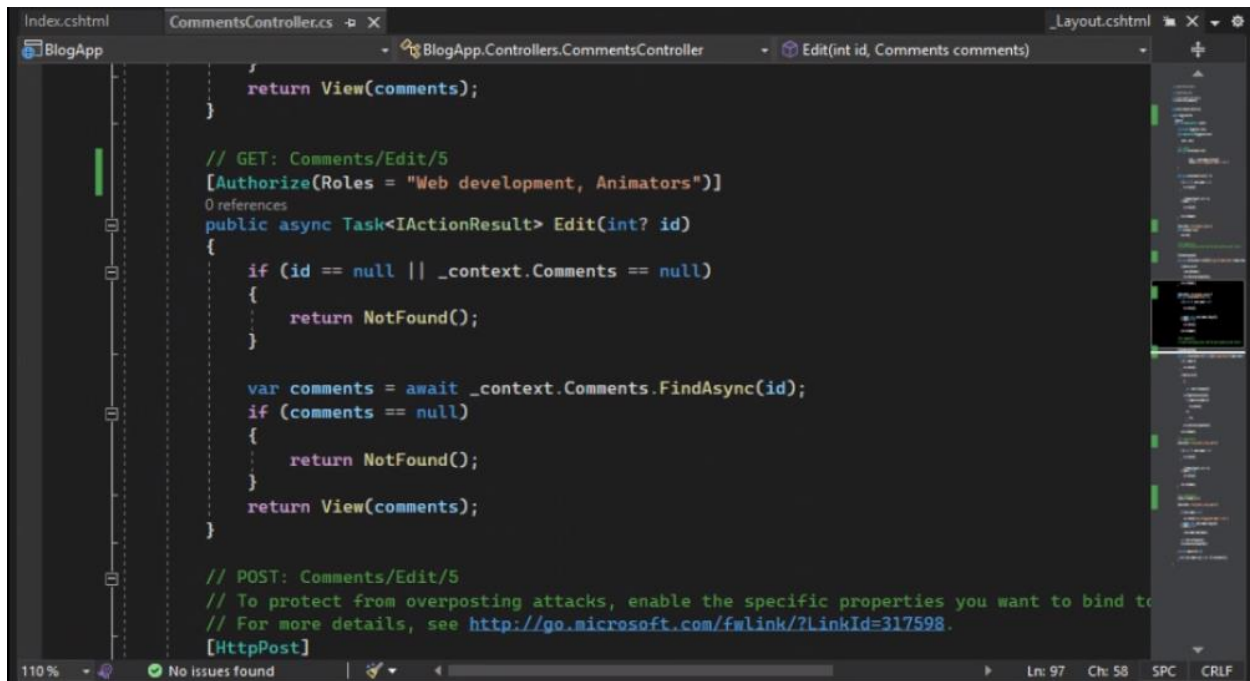
This screenshot shows the Visual Studio IDE with the `CommentsController.cs` file open. The `Create` method is visible, which is an asynchronous action method. It includes authorization attributes for roles "Web development" and "Animators". The method first checks if the model state is valid. If valid, it adds the comment to the context, saves changes, and redirects to the `Index` action. If the model state is not valid, it returns the `View` with the `comments` object.

```
Index.cshtml | CommentsController.cs | _Layout.cshtml
BlogApp | BlogApp.Controllers.CommentsController | Edit(int id, Comments comments)

// GET: Comments/Create
[Authorize(Roles = "Web development, Animators")]
0 references
public IActionResult Create()
{
    return View();
}

// POST: Comments/Create
// To protect from overposting attacks, enable the specific properties you want to bind to
// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
[Authorize(Roles = "Web development, Animators")]
0 references
public async Task<IActionResult> Create([Bind("Id,Type,Title,Comment,PostDate")] Comments comments)
{
    if (ModelState.IsValid)
    {
        _context.Add(comments);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    return View(comments);
}
```

110 % | No issues found | Ln: 97 | Ch: 58 | SPC | CRLF



This screenshot shows the Visual Studio IDE with the `CommentsController.cs` file open. The `Edit` method is visible, which is an asynchronous action method. It includes authorization attributes for roles "Web development" and "Animators". The method first checks if the `id` is null or if the `comments` object is null. If either is null, it returns `NotFound()`. Then, it finds the comment by `id`. If the comment is null, it returns `NotFound()`. Finally, it returns the `View` with the `comments` object.

```
Index.cshtml | CommentsController.cs | _Layout.cshtml
BlogApp | BlogApp.Controllers.CommentsController | Edit(int id, Comments comments)

return View(comments);
}

// GET: Comments/Edit/5
[Authorize(Roles = "Web development, Animators")]
0 references
public async Task<IActionResult> Edit(int? id)
{
    if (id == null || _context.Comments == null)
    {
        return NotFound();
    }

    var comments = await _context.Comments.FindAsync(id);
    if (comments == null)
    {
        return NotFound();
    }
    return View(comments);
}

// POST: Comments/Edit/5
// To protect from overposting attacks, enable the specific properties you want to bind to
// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
```

110 % | No issues found | Ln: 97 | Ch: 58 | SPC | CRLF

The implementation for the authorization for forum access is shown below.

```
Index.cshtml | CommentsController.cs | BlogApp | BlogApp.Controllers.CommentsController | Edit(int id, Comments comments) | _Layout.cshtml
[HttpPost]
[ValidateAntiForgeryToken]
[Authorize(Roles = "Web development, Animators")]
0 references
public async Task<IActionResult> Edit(int id, [Bind("Id,Comment,PostDate")] Comments comments)
{
    if (id != comments.Id)
    {
        return NotFound();
    }

    if (ModelState.IsValid)
    {
        try
        {
            _context.Update(comments);
            await _context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
            if (!CommentsExists(comments.Id))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }
    }

    return RedirectToAction(nameof(Index));
}

// GET: Comments/Delete/5
[Authorize(Roles = "Faculty web dev, Faculty animators")]
0 references
public async Task<IActionResult> Delete(int? id)
{
    if (id == null || _context.Comments == null)
    {
        return NotFound();
    }

    var comments = await _context.Comments
        .FirstOrDefaultAsync(m => m.Id == id);

    if (comments == null)
    {
        return NotFound();
    }

    _context.Remove(comments);
    await _context.SaveChangesAsync();

    return RedirectToAction(nameof(Index));
}
```

```
Index.cshtml | CommentsController.cs | BlogApp | BlogApp.Controllers.CommentsController | Edit(int id, Comments comments) | _Layout.cshtml
{
    return NotFound();
}
else
{
    throw;
}
}

return RedirectToAction(nameof(Index));
}

return View(comments);
}

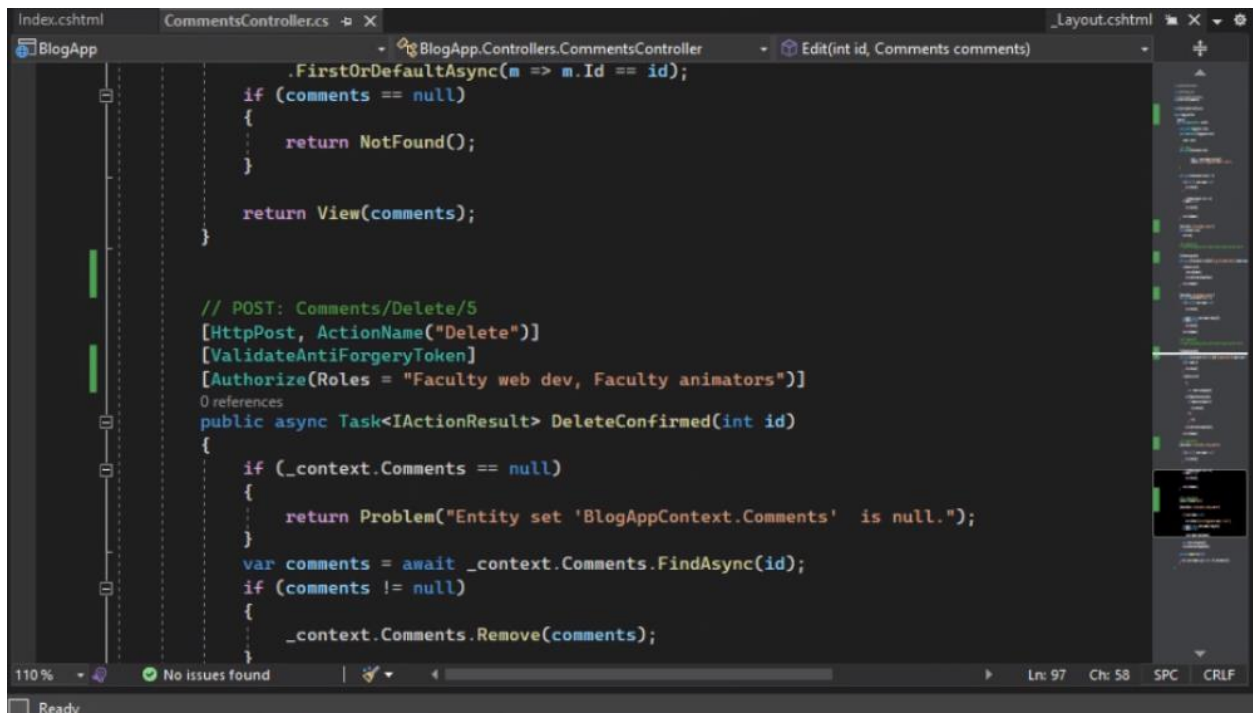
// GET: Comments/Delete/5
[Authorize(Roles = "Faculty web dev, Faculty animators")]
0 references
public async Task<IActionResult> Delete(int? id)
{
    if (id == null || _context.Comments == null)
    {
        return NotFound();
    }

    var comments = await _context.Comments
        .FirstOrDefaultAsync(m => m.Id == id);

    if (comments == null)
    {
        return NotFound();
    }

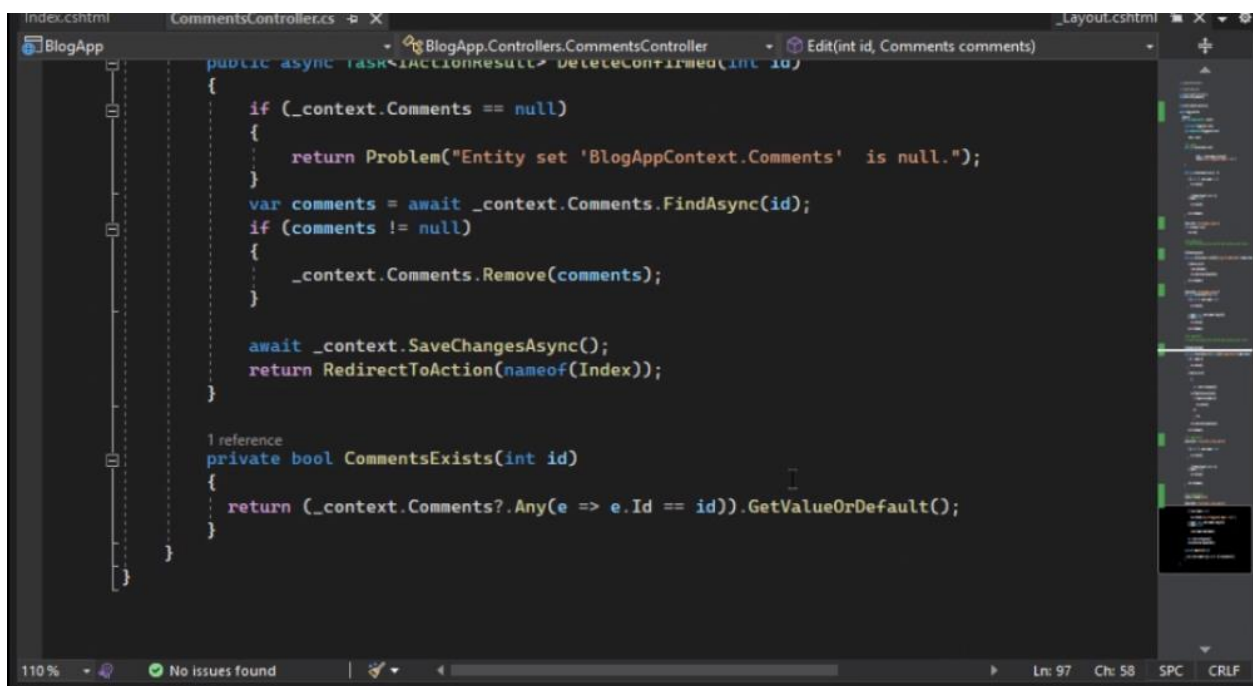
    _context.Remove(comments);
    await _context.SaveChangesAsync();

    return RedirectToAction(nameof(Index));
}
```



```
Index.cshtml | CommentsController.cs | BlogApp | Edit(int id, Comments comments) | _Layout.cshtml
BlogApp
    BlogApp.Controllers.CommentsController
        Edit(int id, Comments comments)
            .FirstOrDefaultAsync(m => m.Id == id);
            if (comments == null)
            {
                return NotFound();
            }
            return View(comments);
        }
        // POST: Comments/Delete/5
        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]
        [Authorize(Roles = "Faculty web dev, Faculty animators")]
        public async Task<IActionResult> DeleteConfirmed(int id)
        {
            if (_context.Comments == null)
            {
                return Problem("Entity set 'BlogAppContext.Comments' is null.");
            }
            var comments = await _context.Comments.FindAsync(id);
            if (comments != null)
            {
                _context.Comments.Remove(comments);
            }
        }
    }
}

110 % | No issues found | Ln: 97 Ch: 58 SPC CRLF
Ready
```



```
Index.cshtml | CommentsController.cs | BlogApp | Edit(int id, Comments comments) | _Layout.cshtml
BlogApp
    BlogApp.Controllers.CommentsController
        Edit(int id, Comments comments)
            .FirstOrDefaultAsync(m => m.Id == id);
            if (comments == null)
            {
                return NotFound();
            }
            return View(comments);
        }
        // POST: Comments/Delete/5
        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]
        [Authorize(Roles = "Faculty web dev, Faculty animators")]
        public async Task<IActionResult> DeleteConfirmed(int id)
        {
            if (_context.Comments == null)
            {
                return Problem("Entity set 'BlogAppContext.Comments' is null.");
            }
            var comments = await _context.Comments.FindAsync(id);
            if (comments != null)
            {
                _context.Comments.Remove(comments);
            }

            await _context.SaveChangesAsync();
            return RedirectToAction(nameof(Index));
        }
        1 reference
        private bool CommentsExists(int id)
        {
            return (_context.Comments?.Any(e => e.Id == id)).GetValueOrDefault();
        }
    }
}

110 % | No issues found | Ln: 97 Ch: 58 SPC CRLF
```

Blogsite Forum Webpage is shown below.

Student's View

Create a blog

[Create New](#)

Title	PostDate	
The Navigating Academia on the Go: The Student's Mobile and Web App Companion	11/15/2023	Edit
Crafting Dreams: The Dynamic World of a Multimedia Arts and Animation Student	11/15/2023	Edit

Faculty's View

Index - BlogApp

Not secure | portal.national-u.edu.ph/Comments

⌵

TheBoysHomeCCITForumsHello, NATIONAL-U\an!

Create a blog

Title	PostDate	
The Navigating Academia on the Go: The Student's Mobile and Web App Companion	11/15/2023	Delete
Crafting Dreams: The Dynamic World of a Multimedia Arts and Animation Student	11/15/2023	Delete

© 2023 - BlogApp - [Privacy](#)