

```

public class CustomAuthorizationFilter : IAuthorizationFilter
{
    public void OnAuthorization(AuthorizationFilterContext context)
    {
        // Verify if the user is logged in
        context.Result = new RedirectToActionResult("Index", "Login", null);
    }
}

```

```

async function fetchTrainings(url) {
    const container = document.getElementById('contentPage');
    container.innerHTML = "";
    await fetchServer(url)
        .then(json => {
            const trainings = [];

            json.forEach(trn => {
                trainings.push(`<div class="col-sm-12 col-md-6 col-lg-6 mb-5">
                    <div class="container d-flex flex-column p-4 mt-3 m-1 card-article dark-shadow">
                        <div class="row mt-auto">
                            
                        </div>
                        <div class="row mt-auto">
                            <span class="pb-3 mb-0 small lh-sm">${trn.name}</span>
                            <p class="pb-3 mb-0 small lh-sm card-bottom">${trn.description}</p>
                        </div>
                        <div class="row mt-auto">${getContentMedia(trn.connect)}</div>
                        <div class="row mt-auto">
                            <a class="card-link" href="${trn.url}" target="_blank">
                                <span>view page</span>
                            </a>
                        </div>
                    </div>
                </div>`);
            });
            document.getElementById('contentPage').innerHTML = `<div class="row">${trainings.join('')}</div>`;
        })
}

```

```
public class LoginController(ISecurityService service) : Controller
{
    private readonly ISecurityService _service = service;

    public IActionResult Index()
    {
        return View("/Views/Accounts/Login.cshtml");
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> Login(LoginModel model)
    {
        bool result = false;
        if (ModelState.IsValid)
            result = await _service.AuthUserByEmailAsync(new Data.Models.User { Email = model.Email,
            Password = model.Password });

        if (result)
            return RedirectToAction("Index", "Home");

        throw new Exception("Error");
    }
}
```