WEB PROGRAMMING ASP.NET MVC CORE

SHOWING ONLY ALLOWED OPTIONS

```
@model IEnumerable<Books.Models.Book>
@{
    ViewData["Title"] = "Index";
}
<h2>Index</h2>
@if (User.IsInRole("Administrator")) {
    <p>
        <a asp-action="Create">Create New</a>
    </p>
}

@* ... *@
```

REGISTER VIEWMODEL

```
public class RegisterCustomerViewModel {
        [Required][StringLength(128)]
        public string Name { get; set; }

        [Required][StringLength(256)][EmailAddress]
        public string Email { get; set; }

        [Required]
        public string Password { get; set; }

        [Required][Compare("Password", ErrorMessage = "The passwords don't match")]
        public string ConfirmPassword { get; set; }
}
```

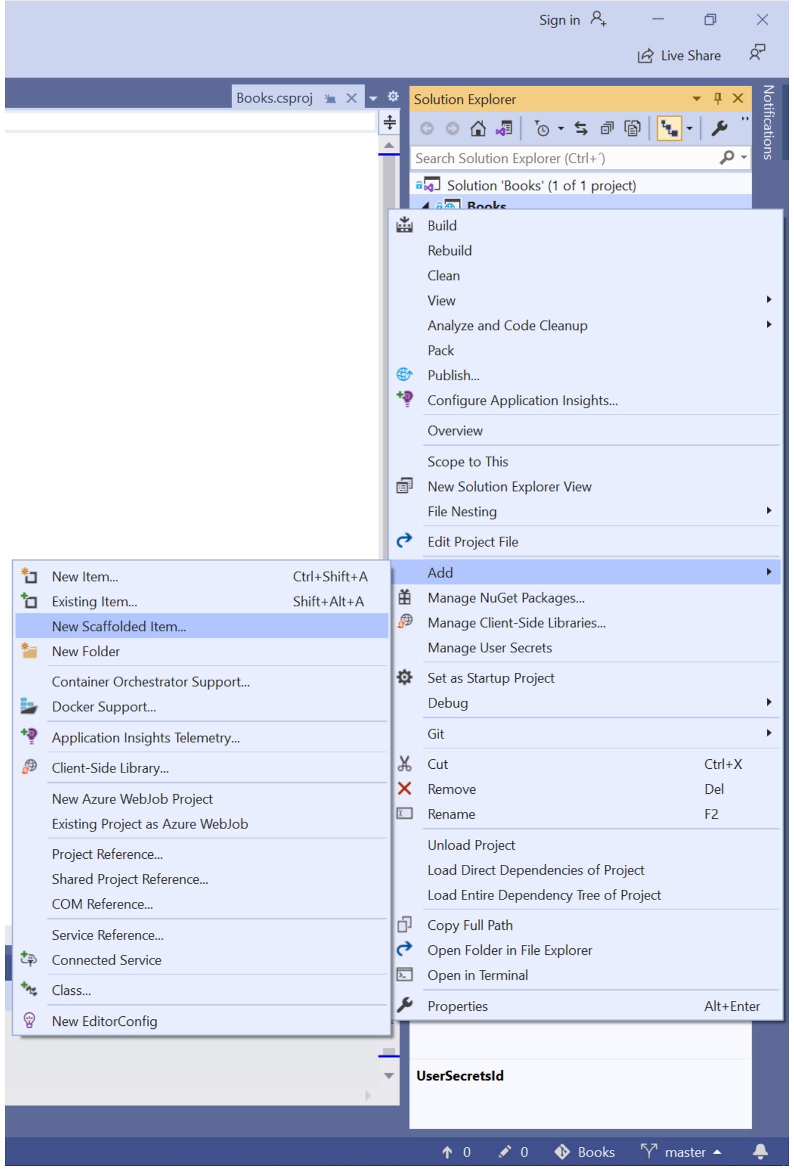
REGISTER CONTROLLER

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Register(RegisterCustomerViewModel customerInfo) {
    if (!ModelState.IsValid) return View(customerInfo);

    string username = customerInfo.Email;
    IdentityUser user = await _userManager.FindByNameAsync(username);
    if (user != null) {
        ModelState.AddModelError("Email", "There is allready a customer with that email.");
        return View(customerInfo);
    }

    user = new IdentityUser(username);
    await _userManager.CreateAsync(user, customerInfo.Password);
    await _userManager.AddToRoleAsync(user, "Customer");

    Customer customer = new Customer { Name = customerInfo.Name,  Email = customerInfo.Email };
    _context.Add(customer);
    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index), "Home");
}
```

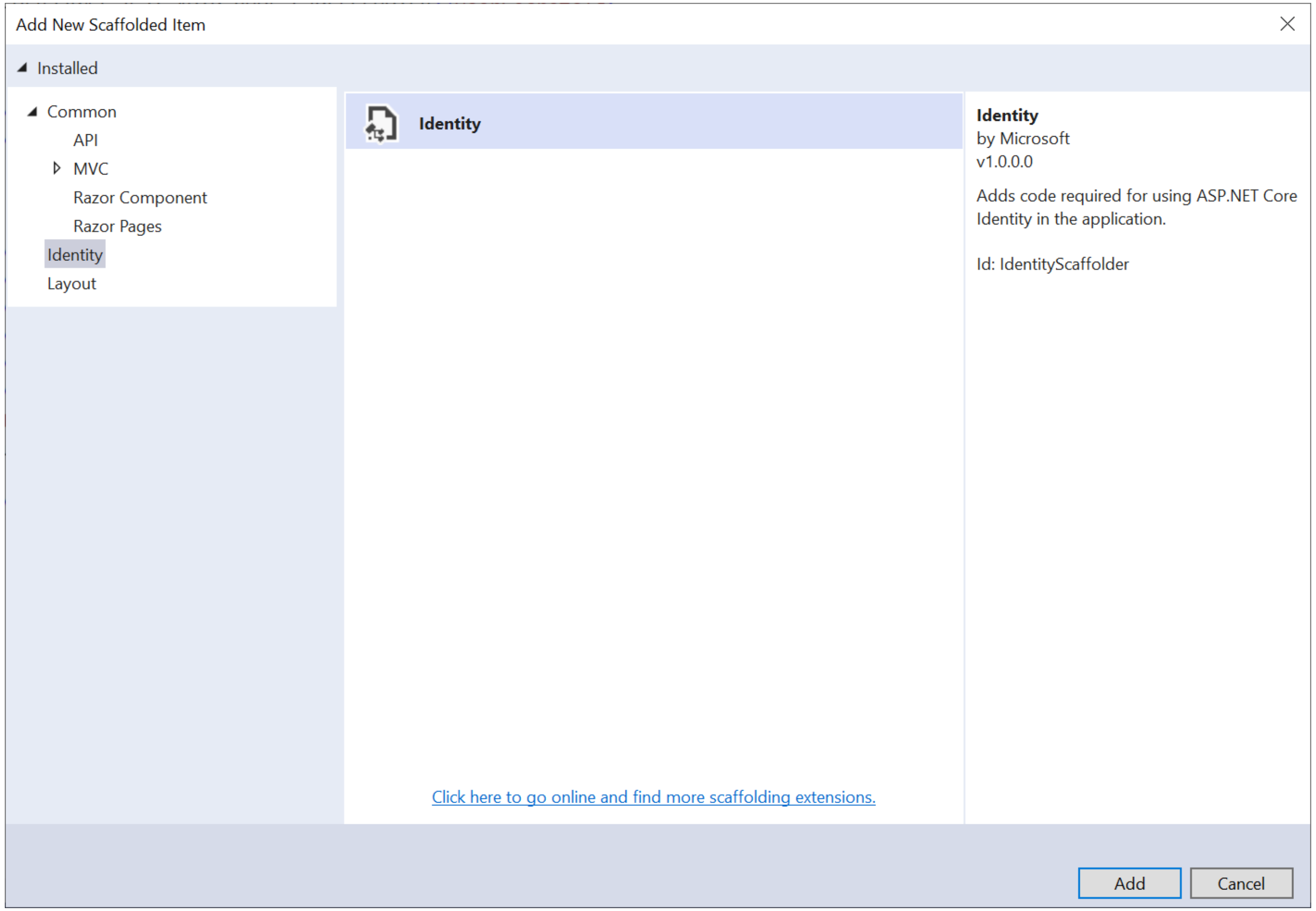
OBTAIN THE ENTITY ASSOCIATED TO THE LOGGED IN USER

```
[Authorize(Roles = "Customer")]
public async Task<IActionResult> EditPersonalData() {
    string email = User.Identity.Name;

    var customer = await _context.Customer.SingleOrDefaultAsync(c => c.Email == email);
    if (customer == null) return NotFound();

    EditLoggedInCustomerViewModel customerInfo = new EditLoggedInCustomerViewModel {
        Name = customer.Name,
        Email = customer.Email
    };

    return View(customerInfo);
}
```

IDENTITY SCAFFOLDING



IDENTITY SCAFFOLDING



Add New Scaffolded Item

Identity
by Microsoft
v1.0.0.0

Adds code required for using ASP.NET Core Identity in the application.

Id: IdentityScaffolder

Click here to go online and find more scaffolding extensions.