

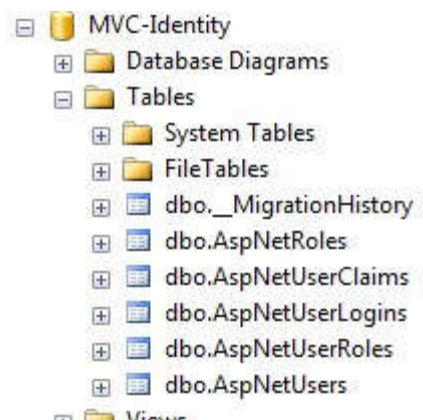
```

public ActionResult AddUserToRole()
{
    AddRolesModel model = new AddRolesModel();
    model.roles.Add("Admin");
    model.roles.Add("Editor");
    model.roles.Add("User");

    // var RoleManager = new RoleManager<IdentityRole>(new
RoleStore<IdentityRole>(new ApplicationDbContext()));
    // RoleManager.Create(new IdentityRole("Admin"));
    // RoleManager.Create(new IdentityRole("User"));
    // RoleManager.Create(new IdentityRole("Editor"));

```

RoleManager класата управува со улогите и AspNetRoles табелата во база ги содржи сите дефинирани улоги



Vo AspNetRoles

Results		Messages
	Id	Name
1	06efeb93-dd37-4482-8fe9-9645dc985099	Admin
2	909e9181-4dda-47e3-9f88-623d1a764b20	Editor
3	a8cb155b-e747-4767-b223-f207b69067bb	User

```

    return View(model);
}

```

UserManager класата управува со корисниците и AspNetUsers табелата во база ги содржи сите корисници

ApplicationUser nasleduva od IdentityUser

```

[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Register(RegisterViewModel model)
{
    if (ModelState.IsValid)
    {
        var user = new ApplicationUser { UserName = model.Email, Email =
model.Email };
        var result = await UserManager.CreateAsync(user, model.Password);
        if (result.Succeeded)
        {
            await SignInManager.SignInAsync(user, isPersistent:false,
rememberBrowser:false);

            // For more information on how to enable account confirmation and
password reset please visit http://go.microsoft.com/fwlink/?LinkID=320771
            // Send an email with this link
            // string code = await
UserManager.GenerateEmailConfirmationTokenAsync(user.Id);
            // var callbackUrl = Url.Action("ConfirmEmail", "Account", new {
userId = user.Id, code = code }, protocol: Request.Url.Scheme);
            // await UserManager.SendEmailAsync(user.Id, "Confirm your account",
"Please confirm your account by clicking <a href=\"" + callbackUrl + "\">here</a>");

            return RedirectToAction("Index", "Home");
        }
        AddErrors(result);
    }
}

public class ApplicationUser : IdentityUser
{
    public async Task<ClaimsIdentity>
GenerateUserIdentityAsync(UserManager<ApplicationUser> manager)
    {
        // Note the authenticationType must match the one defined in
CookieAuthenticationOptions.AuthenticationType
        var userIdentity = await manager.CreateIdentityAsync(this,
DefaultAuthenticationTypes.ApplicationCookie);
        // Add custom user claims here
        return userIdentity;
    }
}

```

so registracija se dodavaat vo AspNetUsers baza

Results		Messages			
	Id	Email	EmailConfirmed	PasswordHash	SecurityStamp
1	d8d97ad3-1434-4be8-8e3b-e30c5dd2423c	alek.krstevski@mail.com	0	AFyHNYc+0/99Q9QVGjP0LxURB6EmY0UZonG0YaDwTbKiLMk...	68c5c345fbc5-47b0-bae6-3880

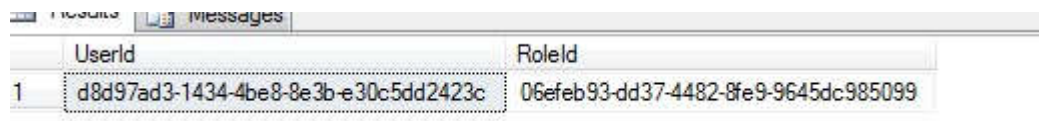
UserManager класата управува со корисниците и AspNetUsers табелата во база ги содржи сите корисници

Методите AddToRole и AddToRoleAsync од класата UserManager овозможуваат на корисник да се додели улога

AspNetUserRoles табелата во база содржи на кој корисник која улога е доделена

```
[HttpPost]
public ActionResult AddUserToRole(AddToRoleModel model)
{
    var user = UserManager.FindByEmail(model.Email);
    UserManager.AddToRole(user.Id, model.selectedRole);
    return RedirectToAction("Index", "Account");
}
```

vo AspNetUserRoles



	Userid	Roleid
1	d8d97ad3-1434-4be8-8e3b-e30c5dd2423c	06efeb93-dd37-4482-8fe9-9645dc985099