# Лабораторная работа 8. Реализация авторизации и аутентификации в приложении ASP.NET Core MVC

В этой работе вы рассмотрите применение системы авторизации и аутентификации в .NET приложениях – ASP.NET Identity. Эта система пришла на смену провайдерам Simple Membership, которые были введены в ASP.NET MVC 4.

## Упражнение 1. Использование типа аутентификации Individual User Accounts

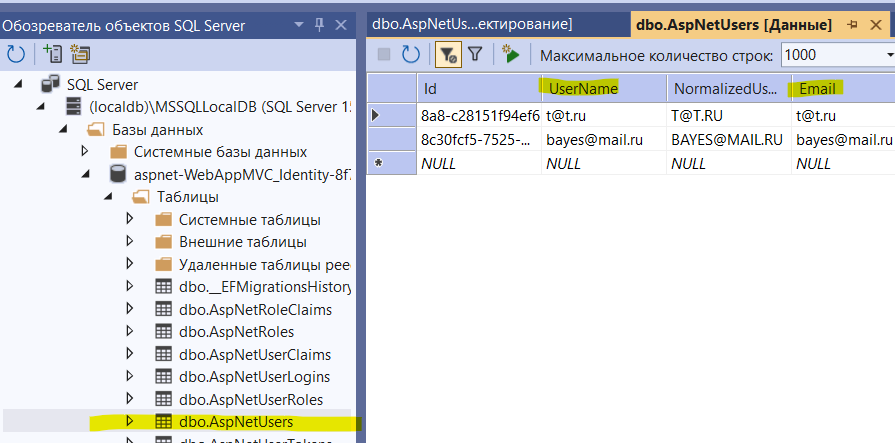
В этом упражнении вы используете в проекте тип аутентификации – **Individual User Accounts**. Созданный проект в работе 5 уже должен иметь всю необходимую для авторизации инфраструктуру: модели, контроллеры, представления.

1. Откройте проект прошлой лабораторной работы.
2. Раскройте узел **Зависимости** и **Пакеты**, проверьте, что там есть ссылки на основные библиотеки, которые содержат необходимые классы для авторизации и аутентификации:

* **Microsoft.AspNetCore.Identity.EntityFrameworkCore**: содержит классы Entity Framework Core, применяющие ASP.NET Core Identity и осуществляющие связь с SQL Server;
* **Microsoft.AspNetCore.Identity**: содержит ряд ключевых интерфейсов ASP.NET Identity.
* Примените миграцию для инициализации базы данных
* Выполните следующую команду в консоли диспетчер пакетов (PMC):

Update-Database -Context 'ApplicationDbContext'

* Проверка возможностей регистрации и входа

1. Запустите приложение.
2. Нажмите на ссылку **Регистрация** (Register). На открывшейся странице создания учетной записи введите свои данные (произвольные) и нажмите кнопку регистрации.
3. Проверьте, что после регистрации логин веб-приложения.
4. Нажмите ссылку **Выйти**, отображающуюся в правом верхнем углу веб-страницы.
5. С помощью обозревателя серверов просмотрите данные регистрации в таблице **AspNetUsers**:
6. 

## Упражнение 2. Ограничение входа для не зарегистрированных пользователей

В этом упражнении вы ограничите доступ к определенным страницам приложения (шаблоны веб-проектов по умолчанию разрешают анонимный доступ).

Чтобы ограничить доступ к представлениям ASP.NET Core MVC, следует ограничить доступ к методу действия, который используется для отображения представления. Для этого платформа предоставляет класс **AuthorizeAttribute**: если метод действия помечается атрибутом AuthorizeAttribute, доступ к этому методу действия ограничивается пользователями, прошедшими проверку подлинности и авторизацию.

* Компоненты авторизации, включая атрибуты AuthorizeAttribute и, AllowAnonymousAttribute находятся в пространстве имен Microsoft.AspNetCore.Authorization.

1. Откройте файл контроллера HomeController и укажите перед методом CreateBid() атрибут [Authorize], теперь создать (отправить) заявку смогут только пользователи, прошедшие проверку подлинности и авторизацию:

[Authorize]

[HttpGet]

public ActionResult CreateBid()

{ ...

1. Запустите приложение и попробуйте без авторизации послать заявку. Вы должны быть перенаправлены на страницу авторизации.
2. Откройте файл контроллера CreditsController и укажите перед классом контроллера атрибут [Authorize], теперь все методы действий в этом контроллере будут ограничены:

[Authorize]

public class CreditsController : Controller

{ ...

1. Запустите приложение и попробуйте без авторизации перейти на страницу *Кредиты*. Вы должны быть перенаправлены на страницу авторизации.