12. Добавление этапов регистрации  
www.youtube.com/watch?v=8ZtInClXe1Q

Суть данной доработки состоит в добавлении этапов создания аккаунта. Создав основу для входа, т.е. добавив email и пароль в БД, следует перенаправить пользователя на страницу ввода метаданных, имени и т.д. Естественно, если процесс был прерван, он должен иметь возможность быть продолженным позднее. Замечу, что в целом это не имеет смысла, лучше объединить всё в одну форму, но поскольку проект учебный, то это очередной proof-of-concept и в реальном проекте я бы так делать не стал.

В модели пользователя создаем поле с текущим этапом регистрации:

public class User

{

**. . .**

public enum REGISTRATION\_STAGE

{

EnteringName, // user needs to enter a name

EnteringMetadata, // City, aboutme, telegram link e.t.c

RegistrationCompleted

}

public REGISTRATION\_STAGE RegistrationStage { get; set; } = REGISTRATION\_STAGE.EnteringName;

}

По-умолчанию задаем первый этап регистрации.

Всего будет 2 дополнительных этапа – ввод имени и ввод метаданных. Создаём две модели дублирующие модель пользователя в нужных полях.  
public class NameModel

{

[DataType(DataType.Text)]

[MinLength(1), MaxLength(50)]

[Required(ErrorMessage = "Please enter a correct user name")]

public string FirstName { get; set; } = null!;

[DataType(DataType.Text)]

[MinLength(1), MaxLength(50)]

[Required(ErrorMessage = "Please enter a correct user name")]

public string LastName { get; set; } = null!;

}

public class MetadataModel

{

[DataType(DataType.Text)]

[MinLength(5), MaxLength(3000)]

[RegularExpression(@"[.]\*[\S]{5}[.]\*")] // at least 5 non-space chars

public string? AboutMe { get; set; } = null!;

[DataType(DataType.Url)]

[MinLength(5), MaxLength(50)]

[RegularExpression(@"[\s]\*[t][.][m][e][/][\S]?")] // \*t.me/?

public string? TelegramLink { get; set; } = null!;

[DataType(DataType.Url)]

[MinLength(5), MaxLength(50)]

[RegularExpression(@"[\s]\*[v][k][.][c][o][m][/][\S]?")] // \*vk.com/?

public string? VkLink { get; set; } = null!;

[DataType(DataType.Text)]

[MinLength(1), MaxLength(50)]

[RegularExpression(@"[.]\*[\S]{5}[.]\*")] // at least 1 non-space chars

public string? City { get; set; } = null!;

[DataType(DataType.PostalCode)]

[MinLength(3), MaxLength(10)]

[RegularExpression(@"[.]\*[\S]{3}[.]\*")] // at least 3 non-space chars

public string? PostalCode { get; set; } = null!;

}

Далее в контроллере, от которого наследуются все контроллеры приложения, пропишем переадресацию на продолжение регистрации в случае, если пользоваться авторизован под аккаунтом, на котором авторизация не завершена:

public override void OnActionExecuting(ActionExecutingContext context)

{

RC.CountRequest(context);

if (RC.IPAddressIsBanned(context)) context.Result = new StatusCodeResult(429);

base.OnActionExecuting(context);

LoadSessionAndAuthedUser();

**RedirectToRegistrationStepIfNeeded(context);**

}

protected void RedirectToRegistrationStepIfNeeded(ActionExecutingContext context)

{

if (

this.AuthedUser != null &&

AuthedUser.RegistrationStage != REGISTRATION\_STAGE.RegistrationCompleted)

{

var act = context.RouteData.Values["Action"];

var ctr = context.RouteData.Values["controller"];

if (AuthedUser.RegistrationStage == REGISTRATION\_STAGE.EnteringName)

{

if (

act is null || ctr is null

|| !act.Equals("RegistrationStageEnteringName")

|| !ctr.Equals("User")){

context.Result

= **RedirectToAction**("RegistrationStageEnteringName", "User");

}

}

if (AuthedUser.RegistrationStage == REGISTRATION\_STAGE.EnteringMetadata)

{

if (

act is null || ctr is null

|| !act.Equals("RegistrationStageEnteringMetadata")

|| !ctr.Equals("User")){

context.Result

= **RedirectToAction**("RegistrationStageEnteringMetadata", "User");

}

}

}

}

Выглядит громоздко, но логика, в целом, простая.