using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Data.Entity;

using System.Linq;

using System.Web;

namespace WebApplication61.Models

{

[Table(name:"UserDetails")]

public class User

{

public int userKey { get; set; }

[Column(name:"FirstName")]

[Required(ErrorMessage ="First Name is must")]

[MinLength(10, ErrorMessage ="Min 10 charctaers needed")]

[StringLength(20, ErrorMessage ="Max 20 charc allowed")]

[RegularExpression(pattern: ("^[a-z A-Z]+$"), ErrorMessage = "Only aplabtets allowed")]

public string FirstName { get; set; }

[Required(ErrorMessage = "Last Name is must")]

[MinLength(10, ErrorMessage = "Min 10 charctaers needed")]

[StringLength(20, ErrorMessage = "Max 20 charc allowed")]

[RegularExpression(pattern: ("^[a-zA-Z ]+$"), ErrorMessage = "Only aplabtets allowed")]

public string LastName { get; set; }

[Required]

[AgeMoreThan30]

public int Age { get; set; }

[Required]

public string Password { get; set; }

[Compare("Password", ErrorMessage ="Passwords do not match")]

public string ReTypePassword { get; set; }

[Required]

[CheckDateRangeAttribute]

public DateTime DateOfjoin { get; set; }

}

public class UserDbContext : DbContext

{

public DbSet<User> Users { get; set; }

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

modelBuilder.HasDefaultSchema("users");

modelBuilder.Entity<User>().ToTable("users");

modelBuilder.Entity<User>().HasKey<int>(s => s.userKey);

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace WebApplication61.Models

{

public class CheckDateRangeAttribute :

ValidationAttribute

{

protected override ValidationResult

IsValid(object value, ValidationContext validationContext)

{

if (value != null)

{

DateTime dt = (DateTime)value;

if (dt != null)

{

if (dt >= DateTime.UtcNow)

{

return ValidationResult.Success;

}

}

}

return new ValidationResult(ErrorMessage ?? "Make sure your date is >= than today");

}

}

public class AgeMoreThan30 : ValidationAttribute

{

protected override ValidationResult IsValid(object value, ValidationContext validationContext)

{

if (value != null)

{

int? age = (int)value;

if (age != null)

{

if (age >= 30)

{

return ValidationResult.Success;

}

}

}

return new ValidationResult(ErrorMessage ?? "Age should be more than 30");

}

}

}

Build ur project

Add Controller