Check Data Using [RegularExpression] Annotation Labs

Perform these labs on your own computer using Visual Studio 2022 or later to ensure you understand the lessons presented in the corresponding videos and lectures.

Lab 1: Add a User Class

Right mouse-click on the **EntityClasses** folder and add a new class named **User**.

Replace the entire contents of the new file with the following code.

```
using System.ComponentModel.DataAnnotations;

namespace DataAnnotationsSamples;

public partial class User
{
  public int UserId { get; set; }
  [Display(Name = "Login Id")]
  public string LoginId { get; set; } = string.Empty;
  public string Password { get; set; } = string.Empty;
  public string ConfirmPassword { get; set; } =
string.Empty;
  [Display(Name = "Email Address")]
  public string EmailAddress { get; set; } =
string.Empty;
  [Display(Name = "Phone Number")]
  public string Phone { get; set; } = string.Empty;
}
```

Lab 2: Use [RegularExpression] Attribute to Check for a Valid Email

Add a **[RegularExpression]** attribute to the **EmailAddress** property so it looks like the following.

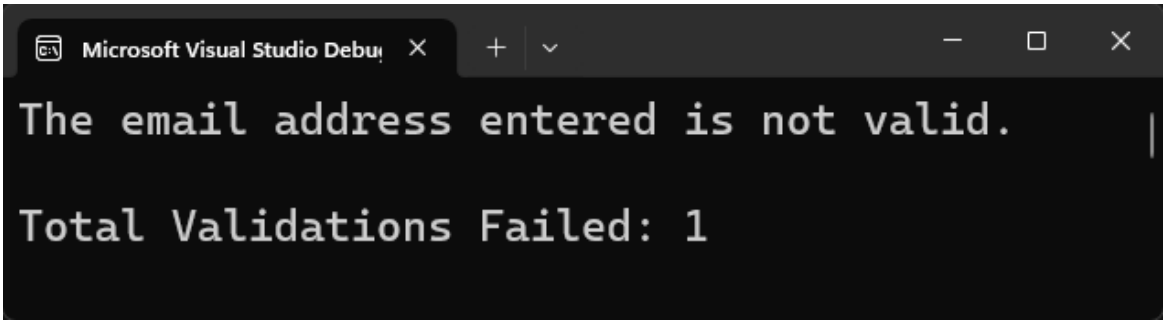
```
[Display(Name = "Email Address")]
[RegularExpression("^\\w+@[a-zA-Z_]+?\\.[a-zA-Z]{2,3}$",
ErrorMessage = "The {0} entered is not valid.")]
public string EmailAddress { get; set; } = string.Empty;
```

Try It Out

Open the **Program.cs** file and remove the *Product entity* declaration, and in its place, put the following code.

```
User entity = new() {
  UserId = 1,
  LoginId = "JoeSmith",
  Password = "Joe!Smith@2024",
  ConfirmPassword = "Joesmith2024",
  EmailAddress = "test!test.com",
  Phone = "xxx-xxx-xxxx"
};
```

Run the application and you should see the error messages as shown in the following screen shot.



Lab 3: Use [RegularExpression] Attribute to Check for a Valid Phone Number

Open the **User.cs** file and add a **[RegularExpression]** attribute to the **Phone** property so it looks like the following.

```
[Display(Name = "Phone Number")]
[RegularExpression("((\\(\\d{3}\\) ?)|(\\d{3}-))?\\d{3}-
\\d{4}", ErrorMessage = "The {0} entered is not valid.
Please use the format (nnn) nnn-nnnn")]
public string Phone { get; set; } = string.Empty;
```

Try It Out

Open the **Program.cs** file and modify the User initialization to make a **valid email** address. Notice that the phone number is invalid.

```
User entity = new() {
  UserId = 1,
  LoginId = "JoeSmith",
  Password = "Joe!Smith@2024",
  ConfirmPassword = "Joesmith2024",
  EmailAddress = "john@smith.com",
  Phone = "xxx-xxx-xxxx"
};
```

Run the application and you should see the error messages as shown in the following screen shot.