

Test (20%)

Test instructions (re-read after finishing):

- DO NOT Share code between students. Cheating will result in 0% and being reported to the college.
- In-class (online), open book test. Max time: 2 hours.
- When finished: Click on Build→ Clean Solution. Close Visual Studio. Zip solution and submit on Lea.
- **IMPORTANT:** Create a new ASP.NET Core project called **TEST-YOUR-STUDENT-ID**
- Store the data to a local database named: **Test2020**

Create a User Account page with front-end and back-end input validation, and store the entry in the database.

- In the nav bar, **add your name** and you **student ID**. **Add your student ID** in the footer as well. Create the 3 links seen after your student id.
- Input validations:
 - All fields are required.
 - Add the same requirements as in the picture.
 - ConfirmPassword must verify that it matches the Password.
 - All error messages appear in **red**.
- The **Reset** button erases all fields in the form.
- Make the default page the Create User Account page.

The screenshot shows a web browser window with the address bar displaying 'localhost:44393/UserAccounts/Create'. The page has a navigation bar with the text 'Test - YOUR NAME - YOUR ID' and two links: 'Create User Account' and 'List all User Accounts'. The main heading is 'Create User Account'. The form contains the following fields and validation messages:

- First Name:** Input field with 'C'. Validation message: 'Must be between 2 and 100 characters'.
- Last Name:** Input field with 'd'. Validation message: 'Must be between 2 and 100 characters'.
- Email:** Input field with 'd'. Validation message: 'The Email field is not a valid e-mail address.'
- Password:** Input field with '..'. Validation message: 'Must be between 5 and 255 characters'.
- Confirm Password:** Input field with '.....'. Validation message: 'The fields Password and Confirm Password should be equals'.

At the bottom of the form are two buttons: 'Create' (blue) and 'Reset' (grey). The footer of the page contains the text '© Test - YOUR STUDENT ID Privacy'.

e) Use any of the (unordered) data annotations below:

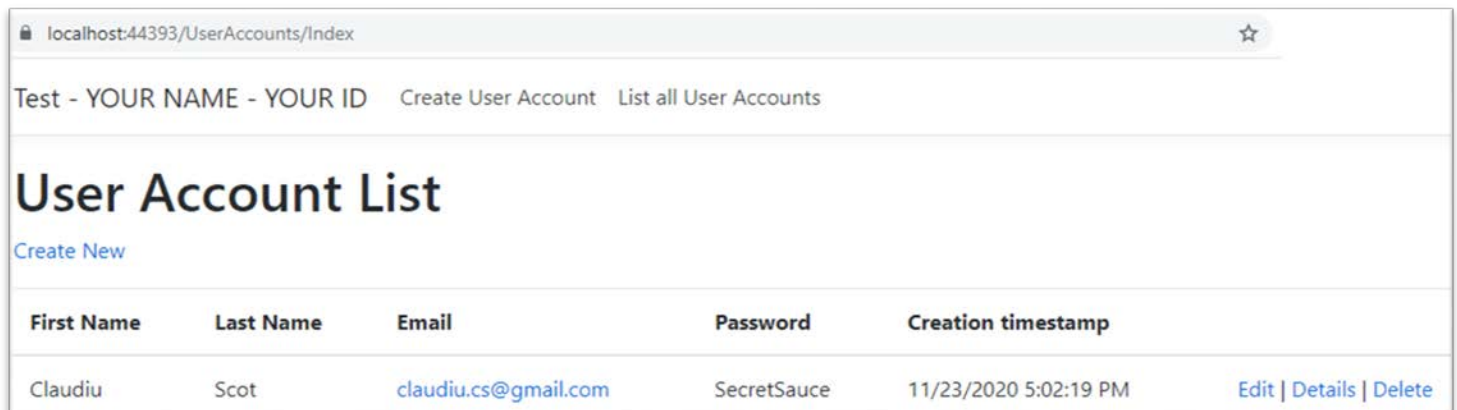
```
[Required(ErrorMessage = "This is required")]
[Display(Name = "The Name")]
[StringLength(200, ErrorMessage = "Must be between 10 and 200 characters", MinimumLength = 10)]
[Compare("Password", ErrorMessage = "Both fields x and Y should be equal")]
[NotMapped]
[Key]
[DataType(DataType.Password)]
[EmailAddress]
```

f) Add data annotation **[NotMapped]** to ConfirmPassword field, to not create the field in the database.

```
[Required]
[DataType(DataType.Password)]
public string Password { get; set; }

[Required]
[DataType(DataType.Password)]
[Compare("Password")]
[NotMapped]
public string ConfirmPassword { get; set; }
```

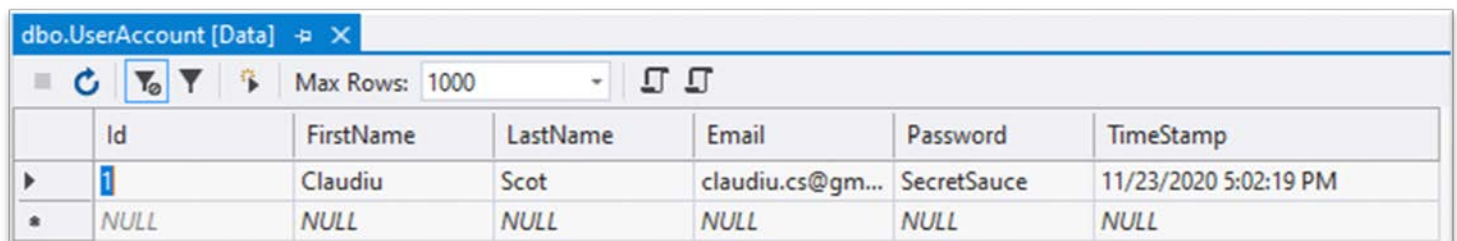
g) Once the form submits, show the User Account List page:



First Name	Last Name	Email	Password	Creation timestamp
Claudiu	Scot	claudiu.cs@gmail.com	SecretSauce	11/23/2020 5:02:19 PM

h) Once the form submits, a user account was created in the local database called *Test2020*

Assume the emails are unique and that there are no duplicates.



	Id	FirstName	LastName	Email	Password	TimeStamp
1	1	Claudiu	Scot	claudiu.cs@gmail.com	SecretSauce	11/23/2020 5:02:19 PM
*	NULL	NULL	NULL	NULL	NULL	NULL

i) When finished:

- a. Click on Build → Clean Solution.
- b. Go to the folder where the Project is located.
- c. Close Visual Studio.
- d. In the folder where the Project is located, delete the **bin** and **obj** folders only.
- e. Zip the solution folder and submit it on Lea.
- f. Re-download the zip.
 - i. Unzip and test what you have submitted.
- j) Enjoy the rest of the day ☺