# Soft Donations Workshop - Part 3

This application is created as a workshop for the [Angular Corse](https://softuni.bg/trainings/2760/angular-january-2020) @ [Software University](https://softuni.bg)

# Task Requirements

At this point the application has **4** **forms**. Now it's the time to edit them so the given information is handled properly as expected. Implement the following **validation**, **notification** and **styling** **rules** for each form.

By default:

* The input fields are **neutral** (neither valid nor invalid). If some of the fields are **touched**, you should determinate the status of validation from the given data inside it
* Each input field is **required**

**Notifications** should be shown as paragraph ("**<p>**") elements **below** each input field or button.

The notification **below each button** should display the **status** of the **form validation**:

* If the status is **invalid**, use - "**You can't continue until you provide a valid data**" as text or something like that. **The color** inside that paragraph should be "**darkred**" and the **button** should be **disabled**.
* If the status is **valid** just **remove** the **notification** and **enable** the **button**.

Check the examples below each form for more clarity.

## Invalid cases

If some of the input fields is invalid. The following **input field** should be styled with:

* "**border:** **1px solid darkred;**"

and **notification** should be shown with information about the validation. The text inside that paragraph also should be **darkred**.

## Valid cases

When the input field is filled with **valid** information. The following input field should be **styled** with:

* "**border**: **1px solid** **darkgreen;**"
* "**font-style: *italic*;**"

and **notification** "*Validation passed*" should be shown.

## Template driven forms

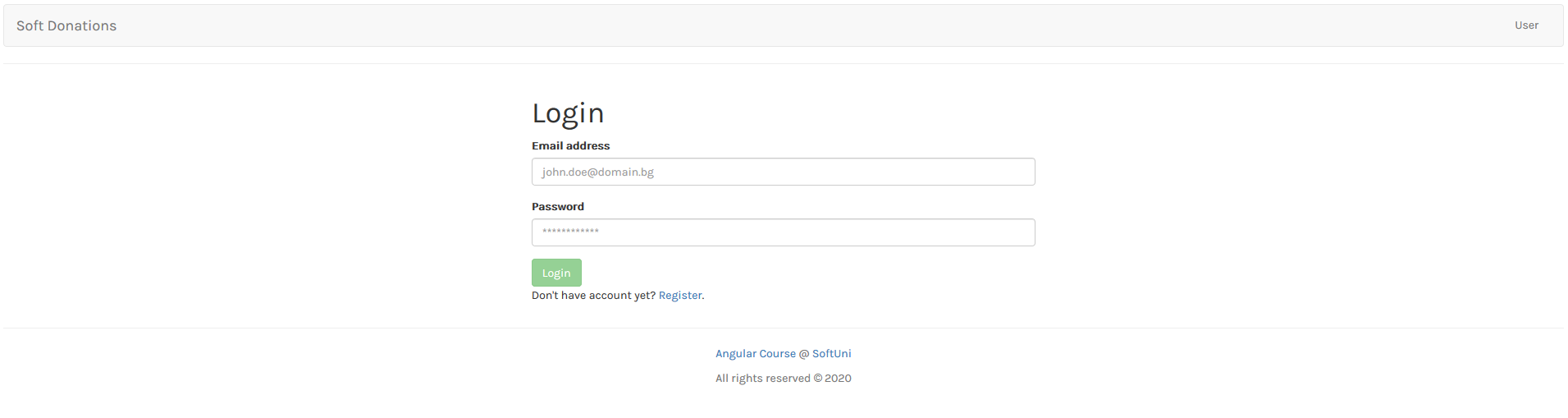
Use the **template** **driven** **approach** to build the **login** and the **register** form.

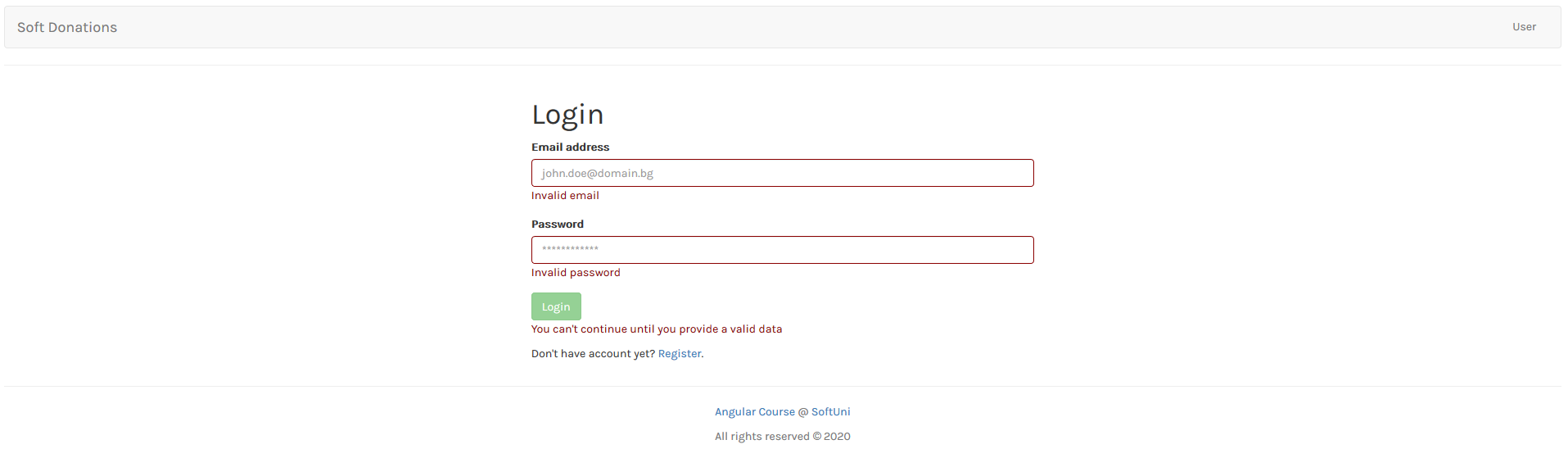
### Login

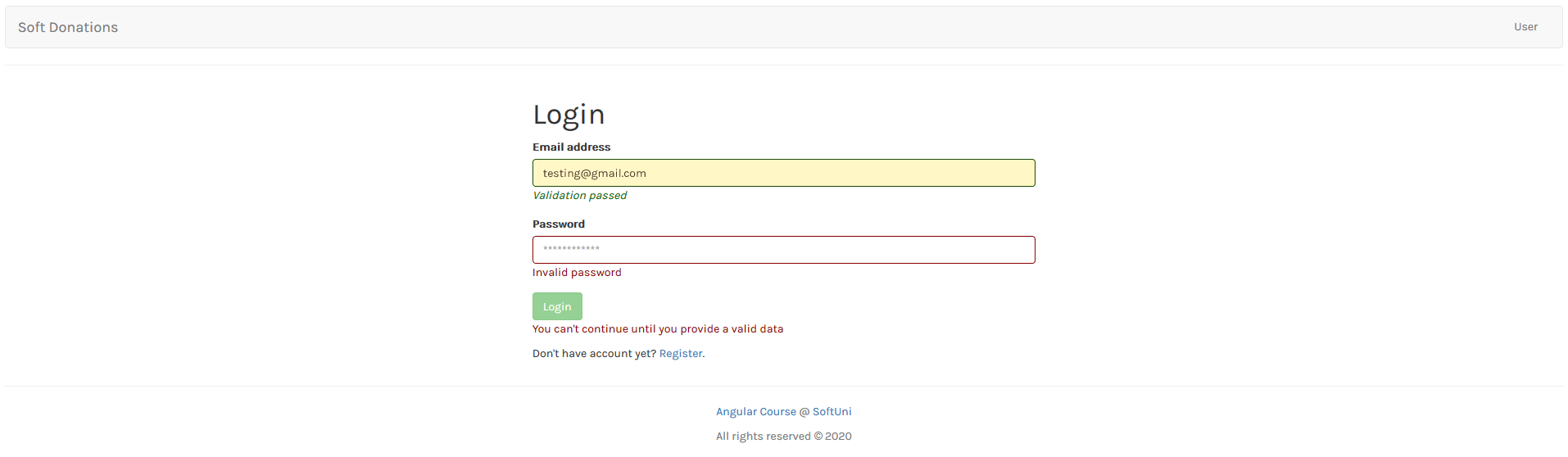
The login form expects as input: **email** and **password**.

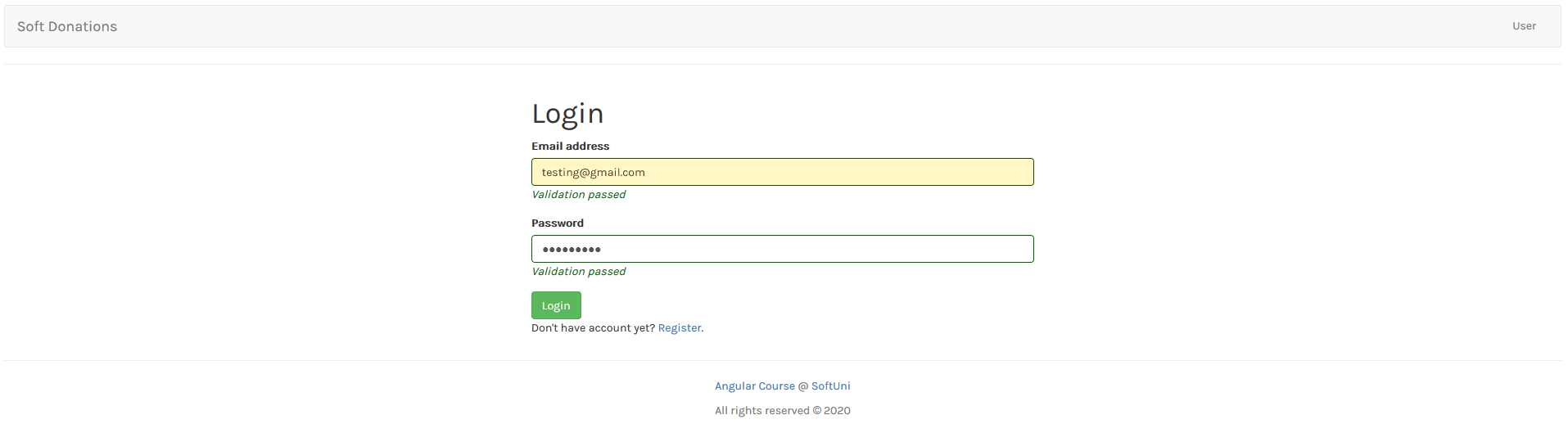
Valid credentials in this case is:

* **Email address** has at least **6 symbols**, after that the symbol of "**@**" is followed. The valid **domain** **name** is only "**gmail**" again followed by "**.**". **Top level** domain can be "**bg**" or "**com**".
  + **Valid** email addresses: [testing@gmail.com](mailto:testing@gmail.com), [testing@gmail.bg](mailto:testing@gmail.bg), [somethingelse@gmail.com](mailto:somethingelse@gmail.com)
  + **Invalid** email addresses: [qwe@gmail.com](mailto:qwe@gmail.com), [123@gmail.bg](mailto:123@gmail.bg), [somethingelse@abv.bg](mailto:somethingelse@abv.bg), [another@abv.com](mailto:another@abv.com)
* **Password** should be at least **6 symbols** as well









### Register

The register form expects as input: **email**, **password** and **repeat password**.

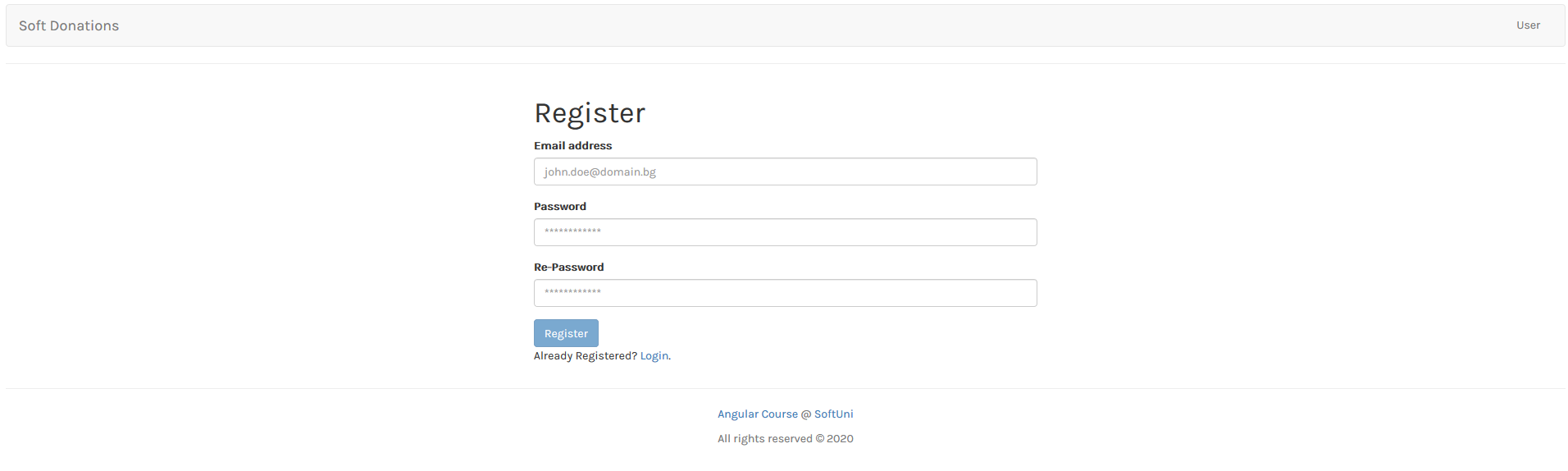
**Email validation** here is the same as the login case:

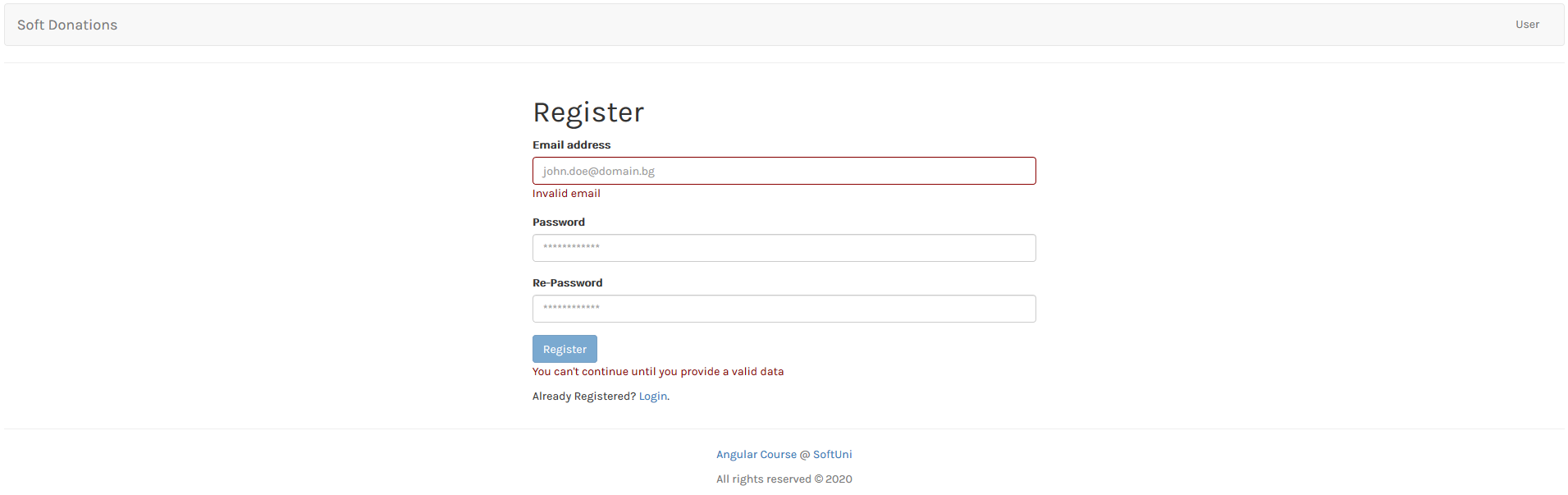
* Should have at least **6 symbols**, after that the symbol of "**@**" is followed. The valid **domain** **name** is only "**gmail**" again followed by "**.**". **Top level** domain can be "**bg**" or "**com**".
  + **Valid** email addresses: [testing@gmail.com](mailto:testing@gmail.com), [testing@gmail.bg](mailto:testing@gmail.bg), [somethingelse@gmail.com](mailto:somethingelse@gmail.com)
  + **Invalid** email addresses: [qwe@gmail.com](mailto:qwe@gmail.com), [123@gmail.bg](mailto:123@gmail.bg), [somethingelse@abv.bg](mailto:somethingelse@abv.bg), [another@abv.com](mailto:another@abv.com)

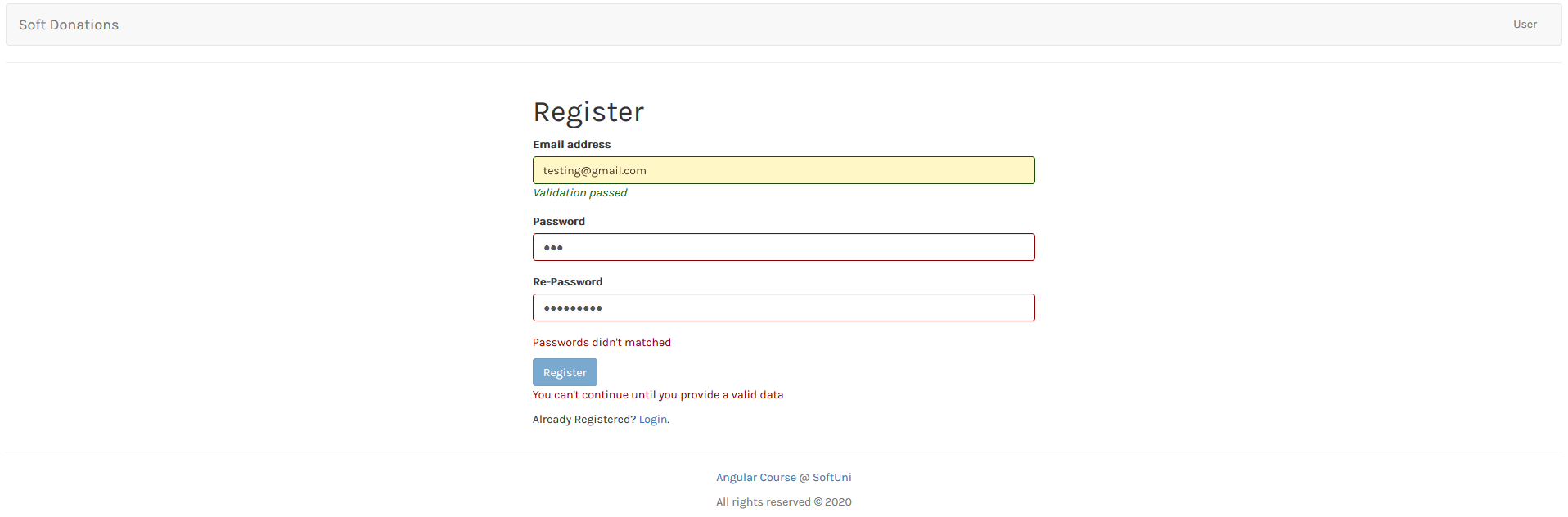
**Both passwords** should be grouped in "**passwords**" group. The validation here is:

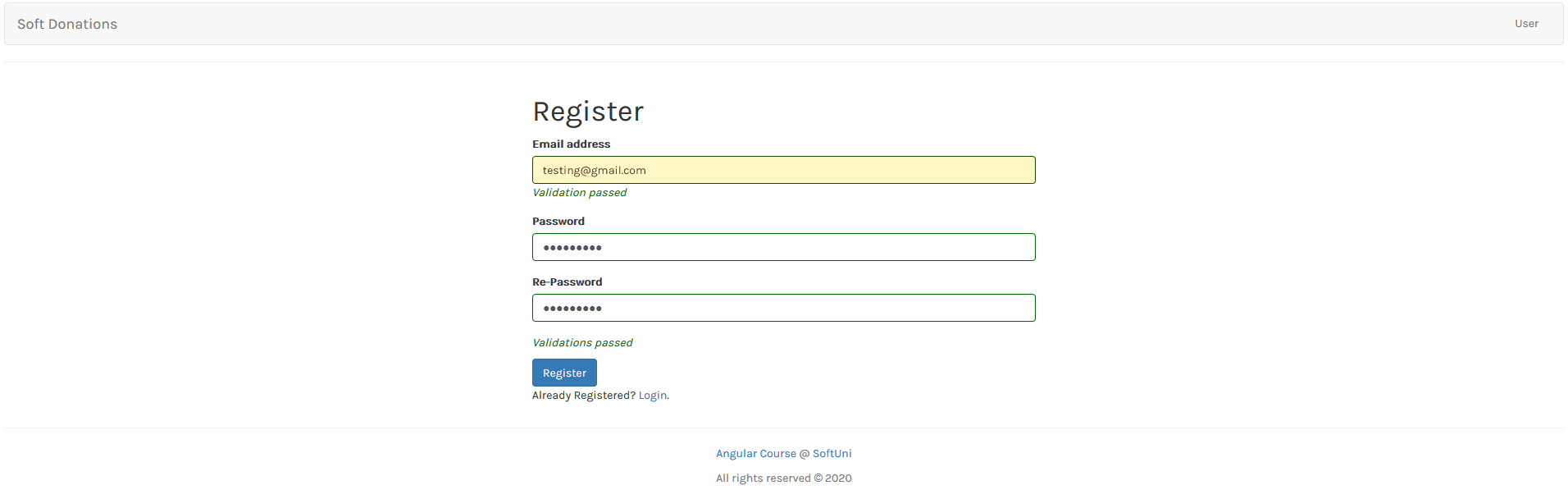
* The given **password** should be at **least 6 characters** (any character allowed)
* The given **repeat password should** be the **same** as the given **password**

**Both fields** are considered **invalid**, if some of them **do not cover** the described validation. If that happen apply the **invalid** **stylization** for **both**. Otherwise the **valid** **stylization** is applied.









## Reactive forms

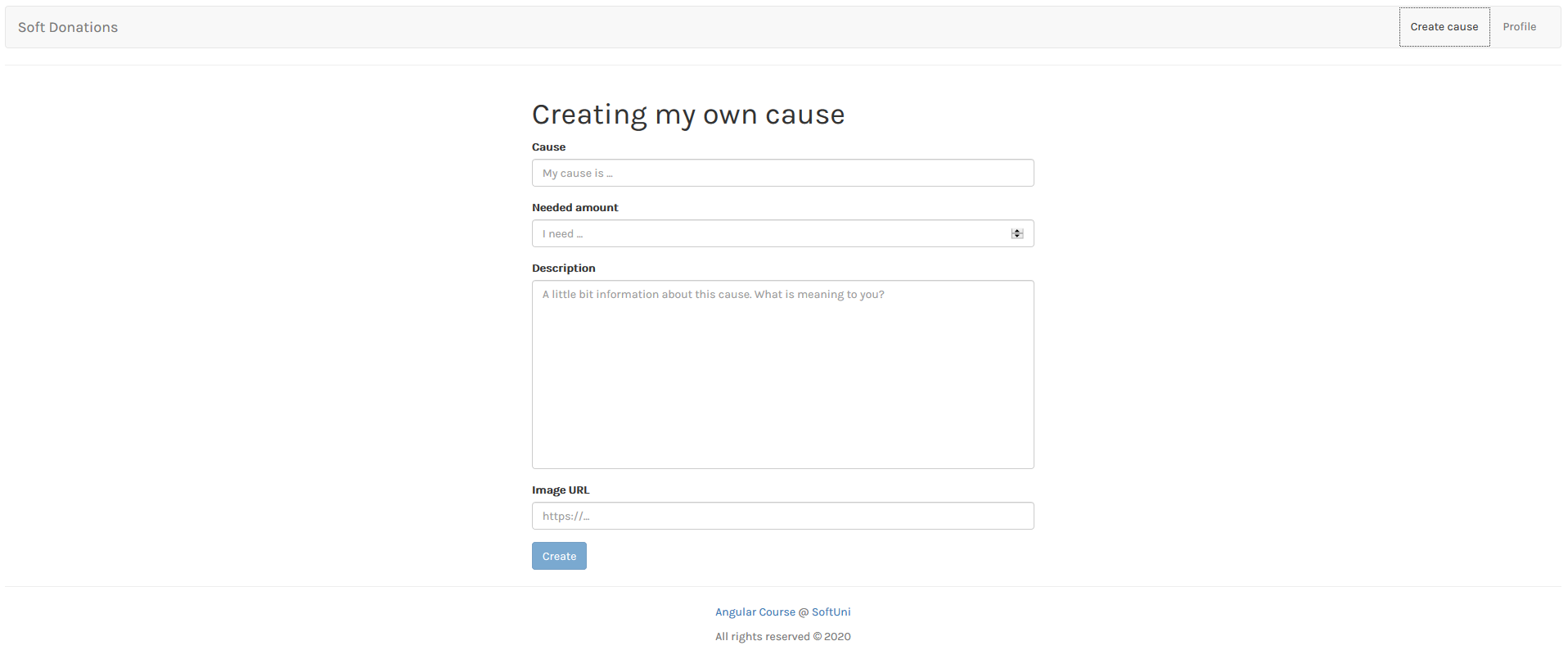
Use the **model driven (reactive)** **approach** to build the **create cause** and the **change user information** form.

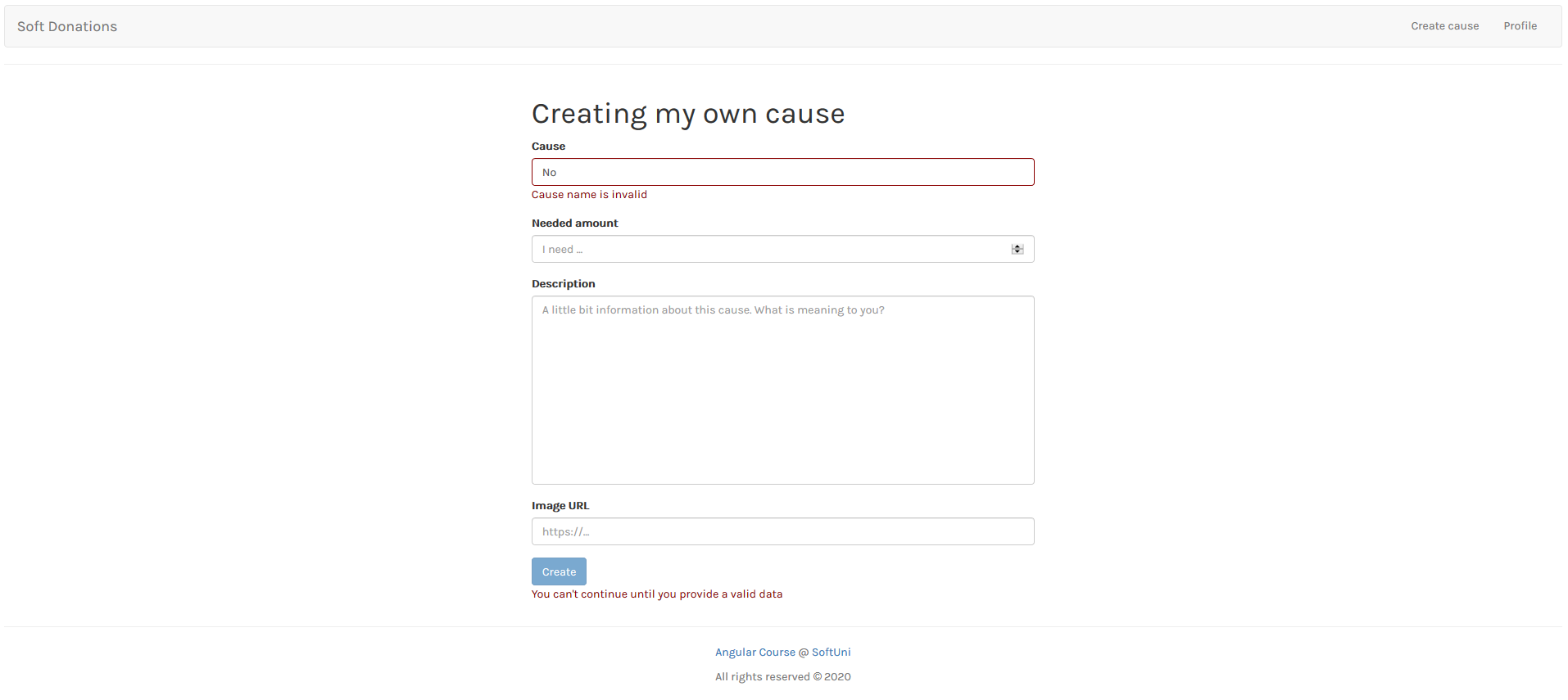
### Create cause

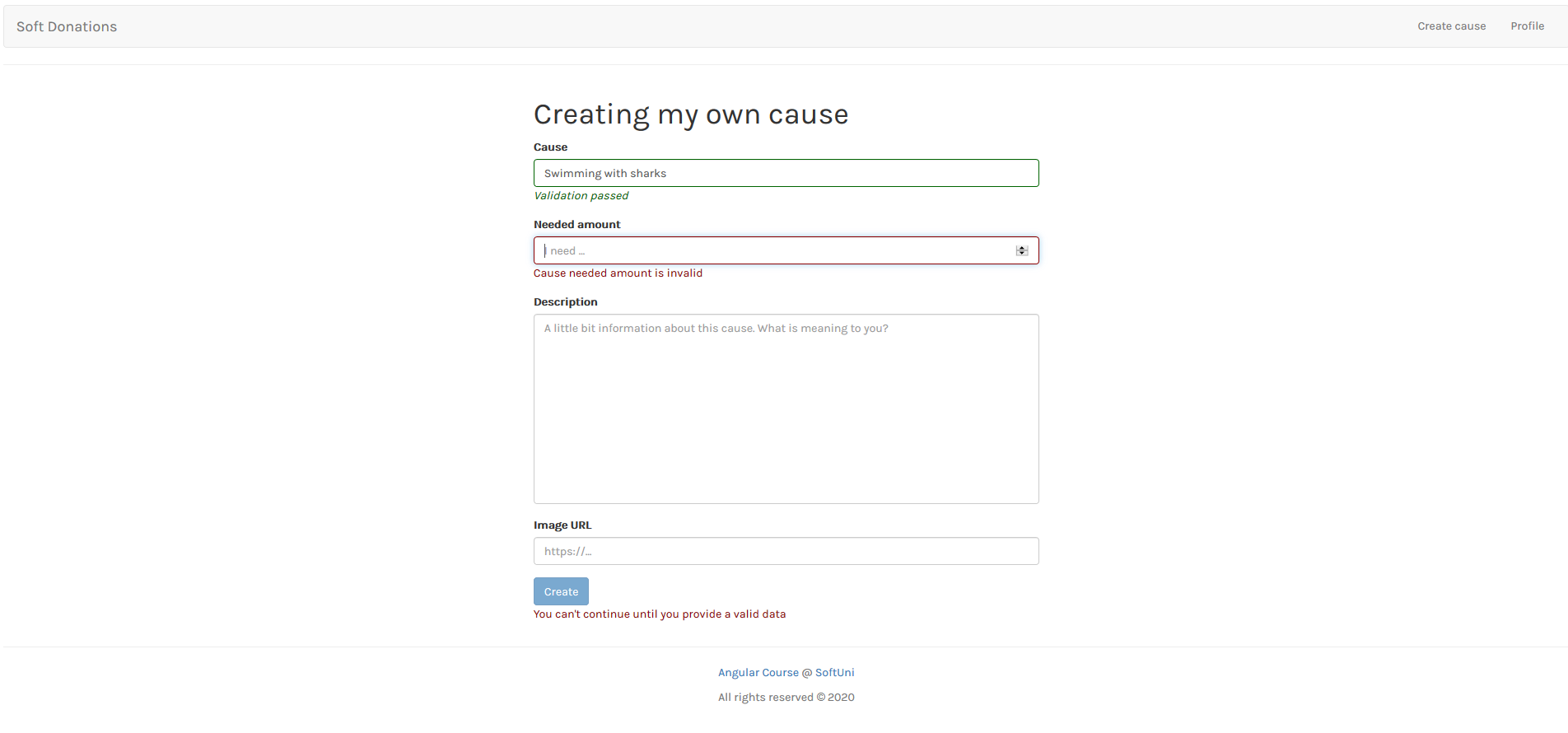
The create case form expects as input: **cause name**, **needed amount**, **description**, **image url**.

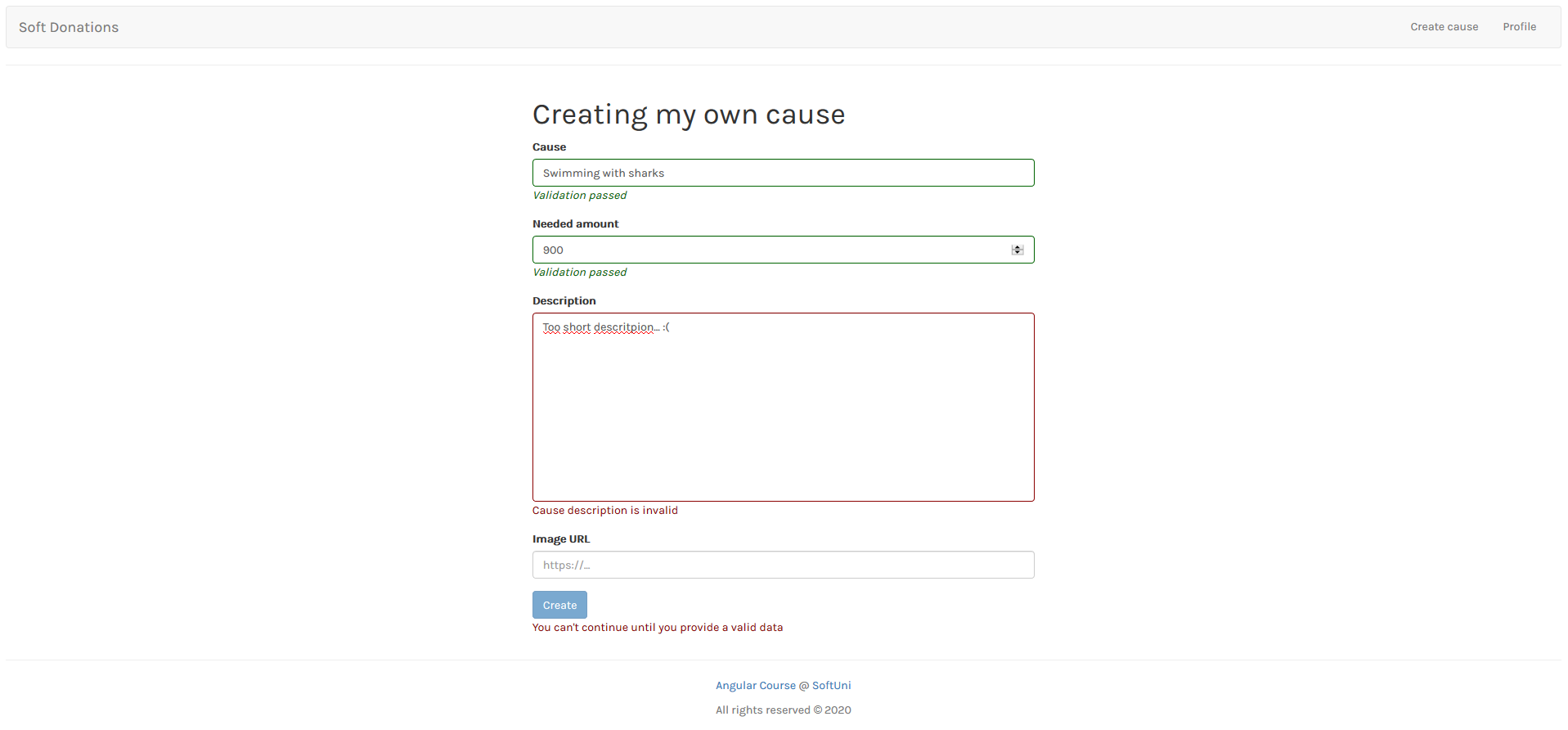
Valid data in this case is:

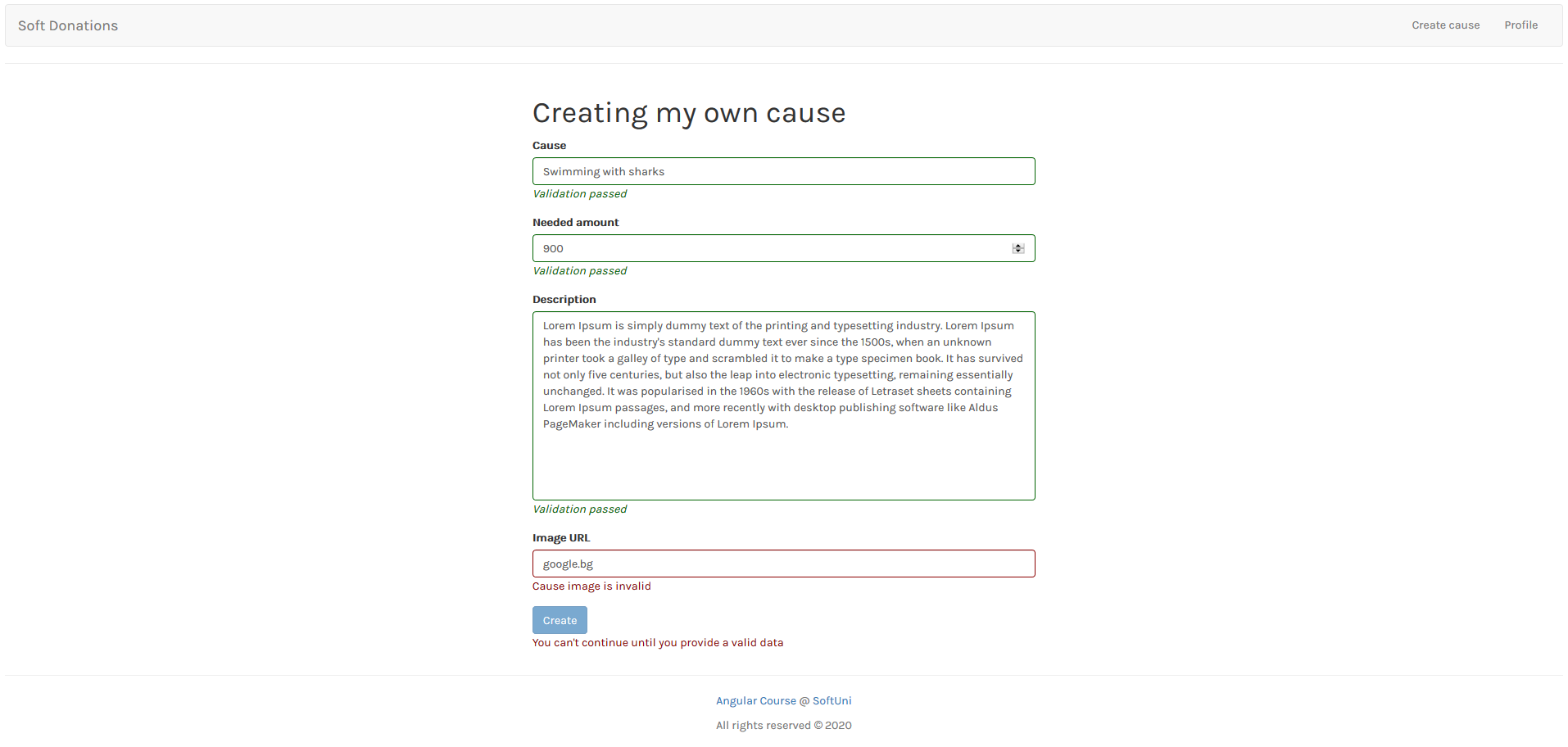
* Cause name - should be at **least 5 characters**
* Needed amount - should be **positive number greater than 100**
* Description - should be at **least 50 characters** long
* Image url - should **starts** with **http** or **https**

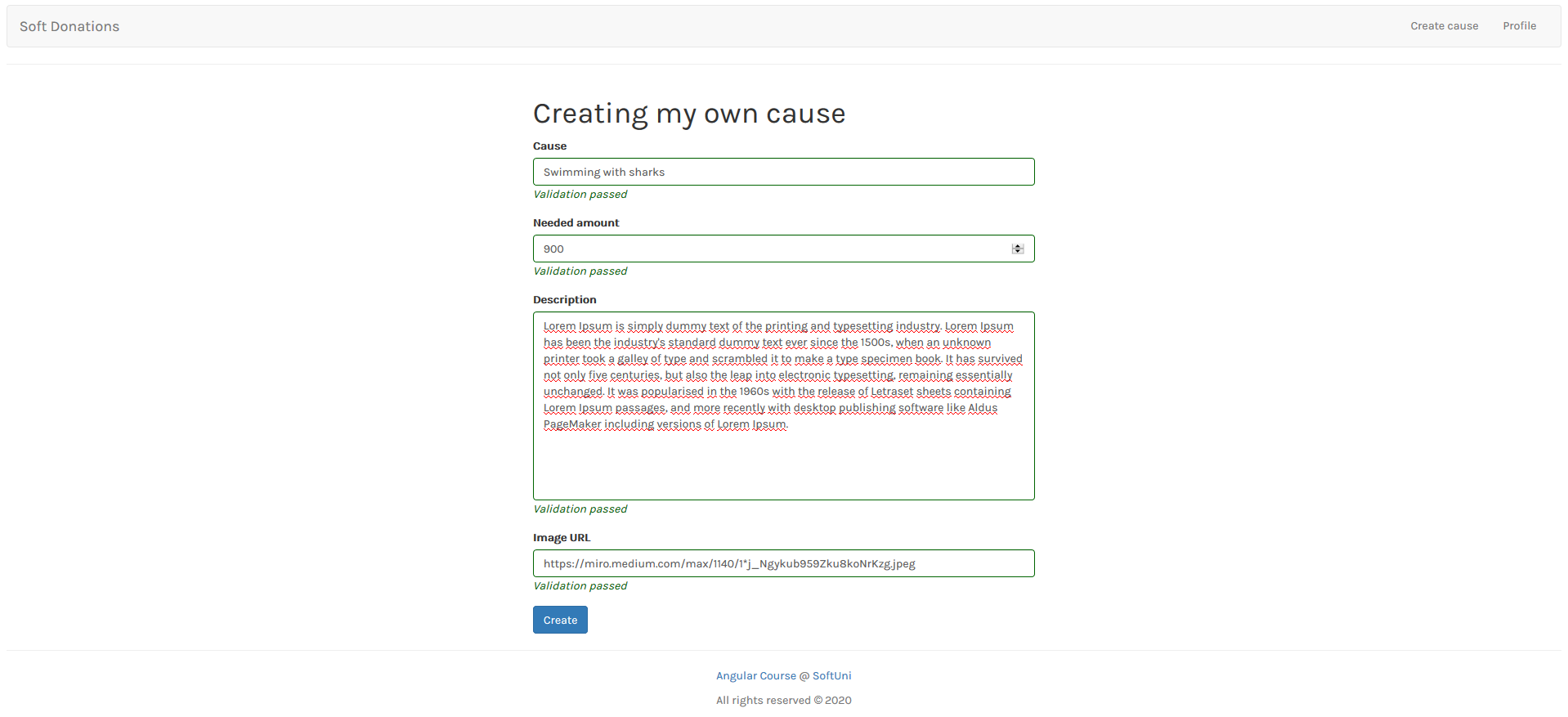












### Change profile information

The change profile information form expects as input: **email, password and repeat password**

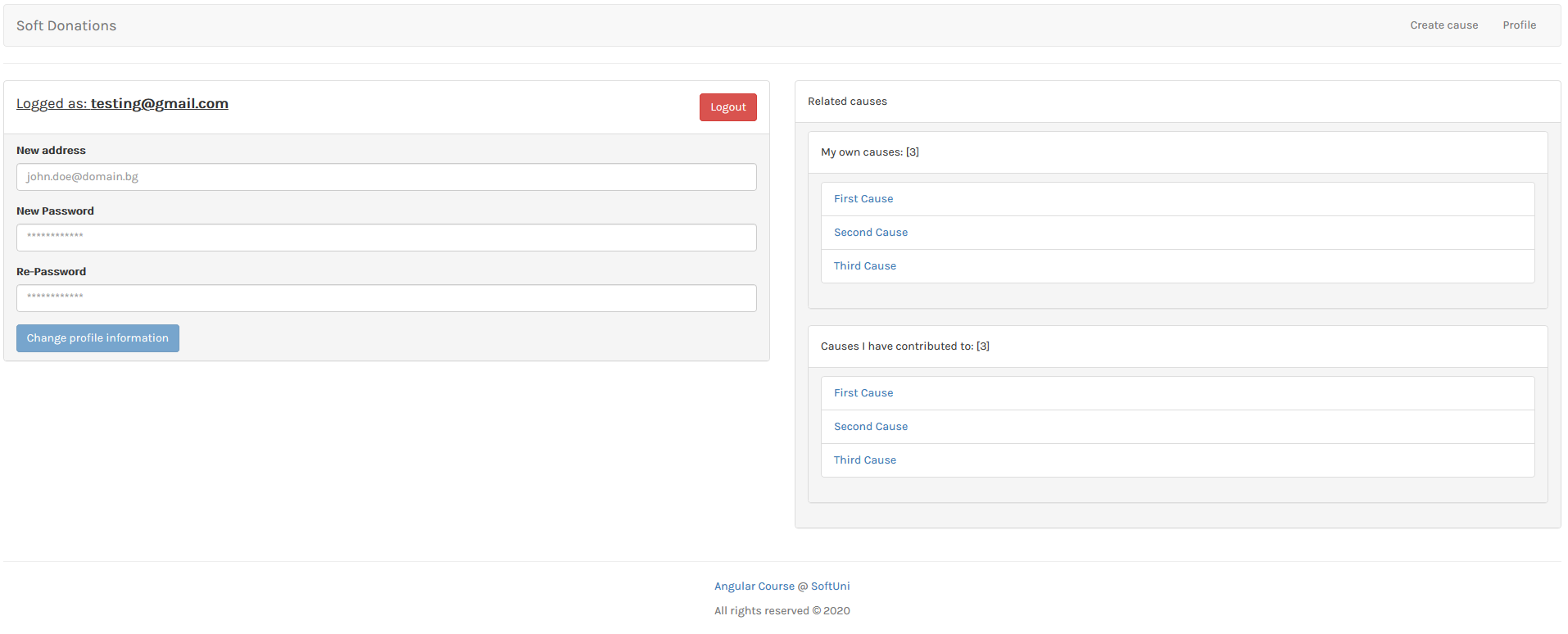
**Email** validation is the same as the login and register cases:

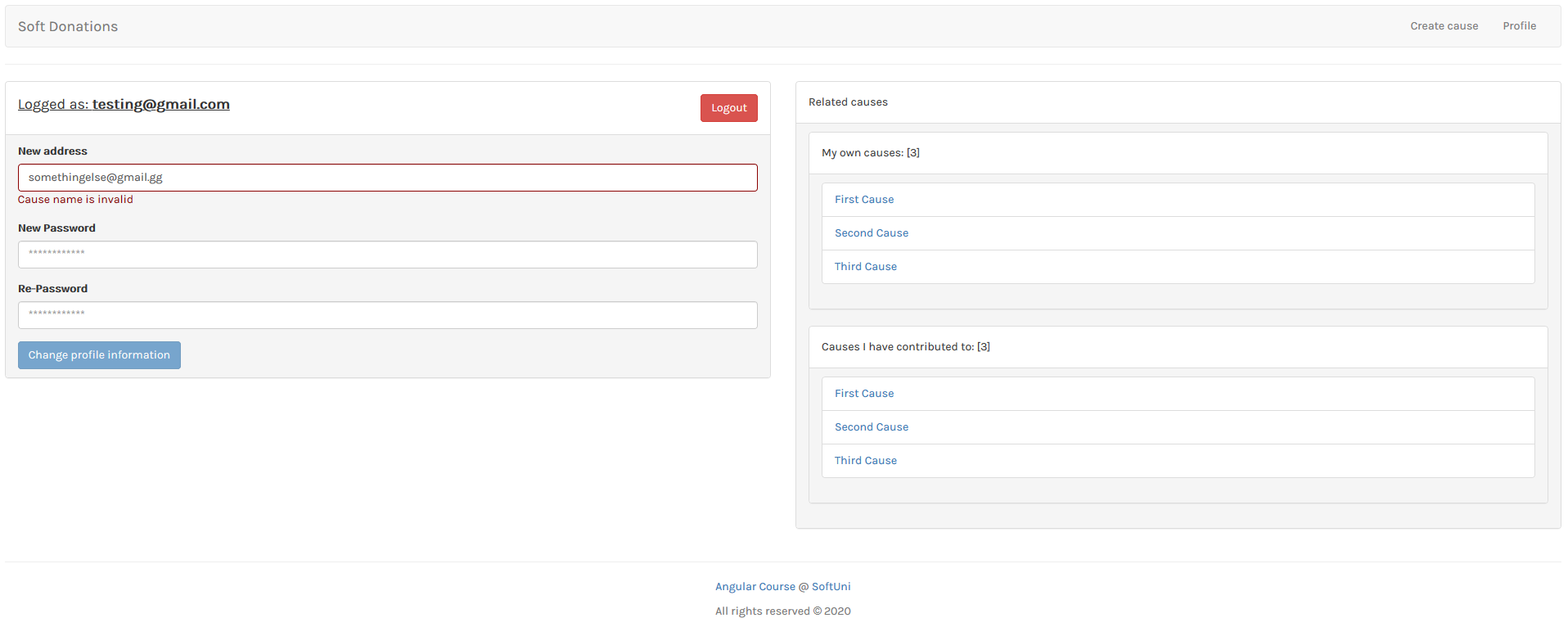
* Should have at least **6 symbols**, after that the symbol of "**@**" is followed. The valid **domain** **name** is only "**gmail**" again followed by "**.**". **Top level** domain can be "**bg**" or "**com**".
  + **Valid** email addresses: [testing@gmail.com](mailto:testing@gmail.com), [testing@gmail.bg](mailto:testing@gmail.bg), [somethingelse@gmail.com](mailto:somethingelse@gmail.com)
  + **Invalid** email addresses: [qwe@gmail.com](mailto:qwe@gmail.com), [123@gmail.bg](mailto:123@gmail.bg), [somethingelse@abv.bg](mailto:somethingelse@abv.bg), [another@abv.com](mailto:another@abv.com)

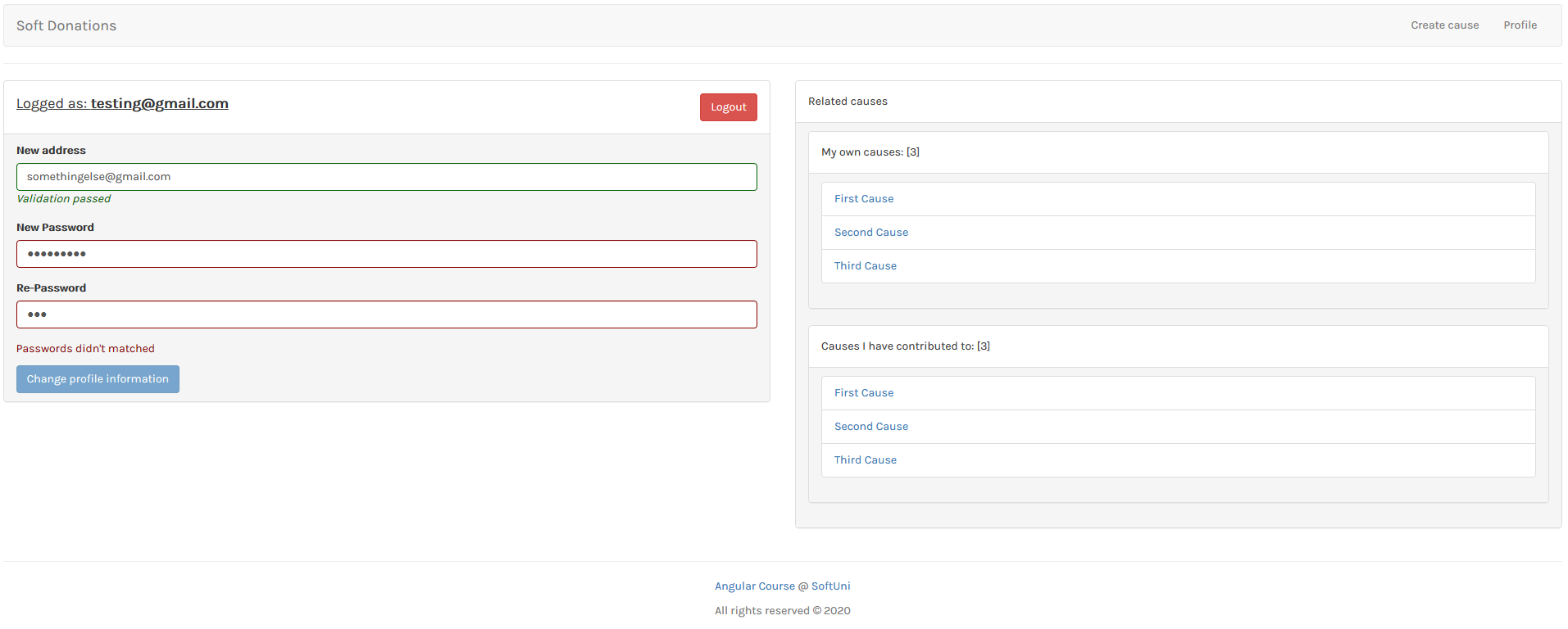
**Both passwords** should be grouped in "**passwords**" group. The validation here is:

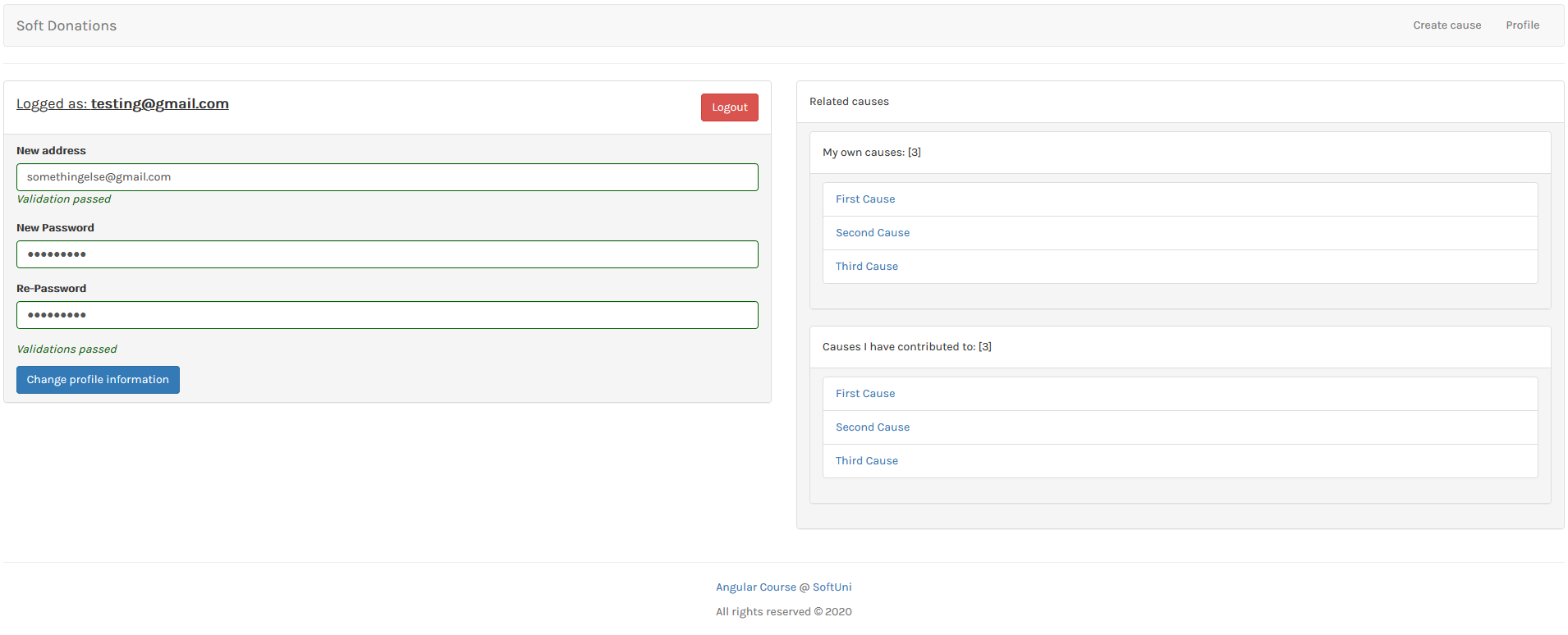
* The given **password** should be at **least 6 characters** (any character allowed)
* The given **repeat password should** be the **same** as the given **password**

**Both fields** are considered **invalid**, if some of them **do not cover** the described validation. If that happen apply the **invalid** **stylization** for **both**. Otherwise the **valid** **stylization** is applied.









## Custom directives

Think about for a second which logic can be isolated in custom directive so you avoid repeating part of the code.

**Hint, Hint: Stylization/Validation c(:**

***GOOD LUCK* 😊**