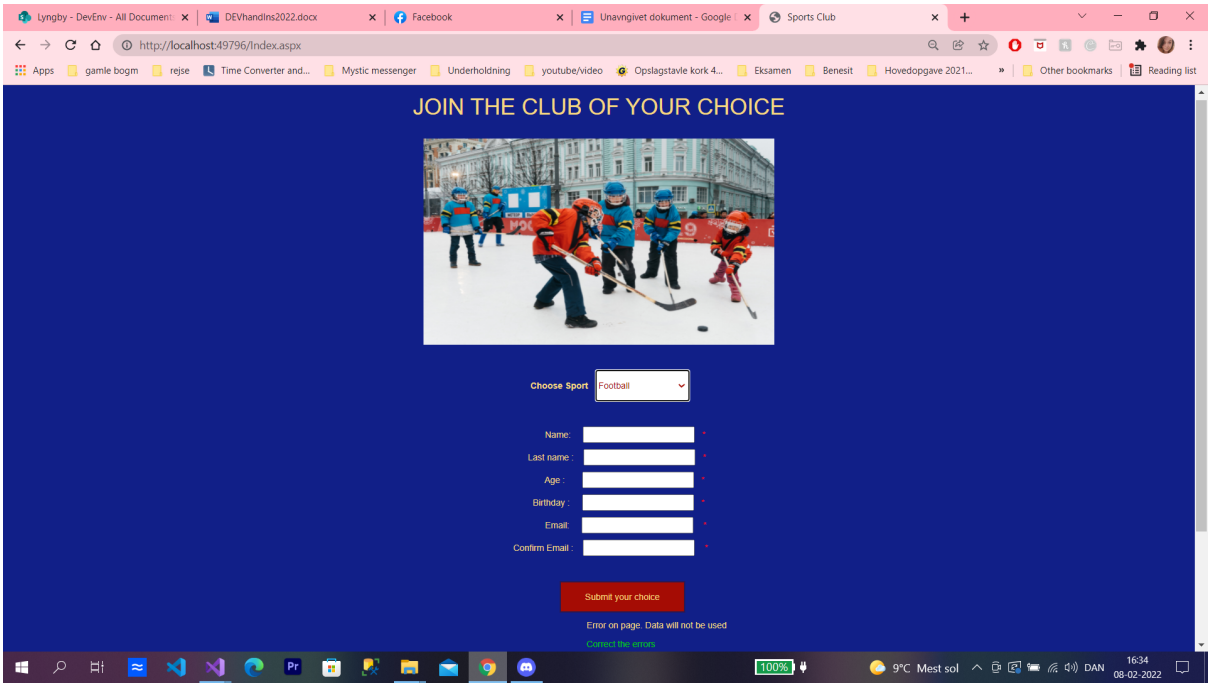
Development Environment 1st Handin

Server Side Validation

Mie Grønkjær Jørgensen
Web Developer 2end semester
cph-mj761@cphbusiness.dk



Getting started

First I created a new project using the Web Application .Net framework with C#, and called the project SportsClub.

Index.aspx

I then created a new webform and called it Index.aspx. This is the main file which I am going to use.

Design mode

I wanted to create a pretty simple design using labels, textboxes and the new validators, which we have just been taught about.
I have used Bootstrap library to make the design aspect easier and I have used it to center a few things on the page.
I chose to have a title centered as well as a picture under it to show the theme of the page.

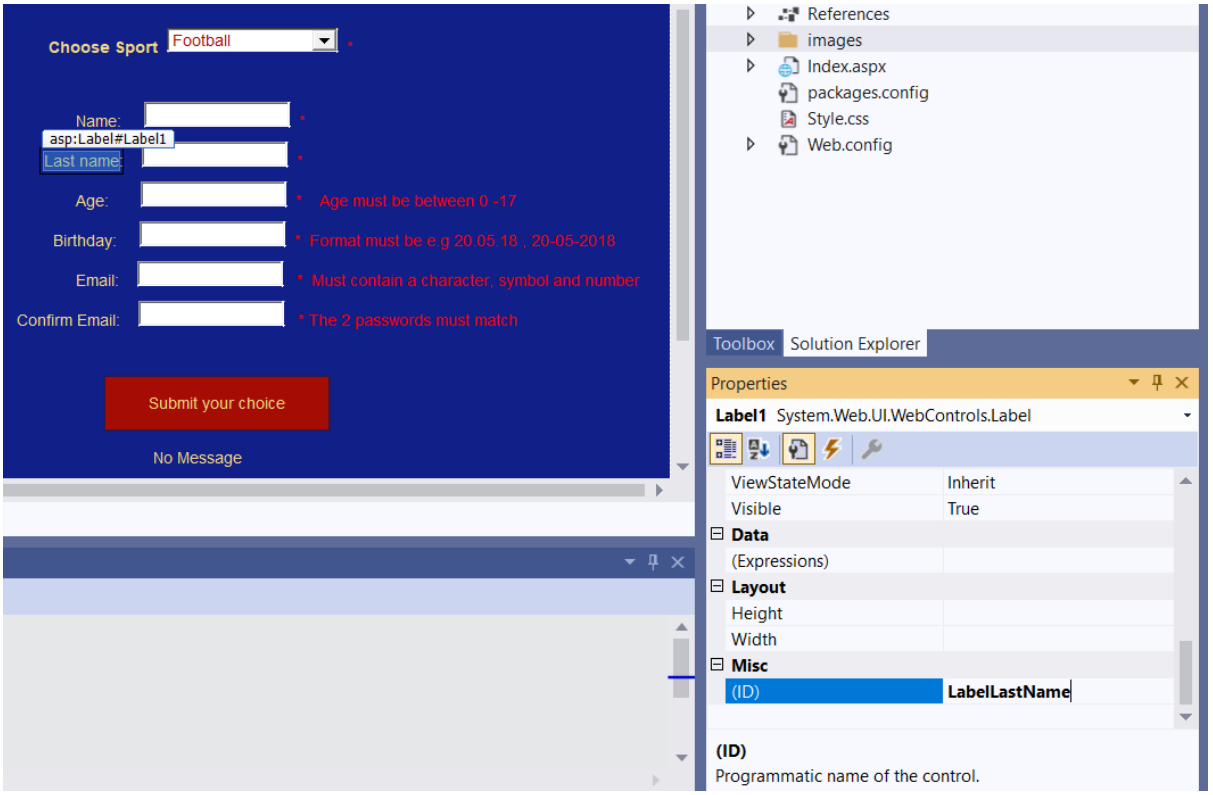
```
<!DOCTYPE html>
   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.
css">
   <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"><
/script>
   <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js
"></script>
   <link rel="stylesheet" href="Style.css"/>
<html xmlns="http://www.w3.org/1999/xhtml">


<head runat="server">
    <title>Sports Club</title>
</head>
<body>
   <div class="container text-center bold">
    <h1> JOIN THE CLUB OF YOUR CHOICE </h1>
   </div>
    <form id="form1" runat="server">
        <div>
        <div class="container text-center">
            <br />
            <asp:Image ID="ImageHockey" runat="server" Height="328px"
ImageUrl="~/images/hockey.jpg" Width="559px" />
            <br />
            <br />
            <br />
```

```
        </div>
        </div>
```

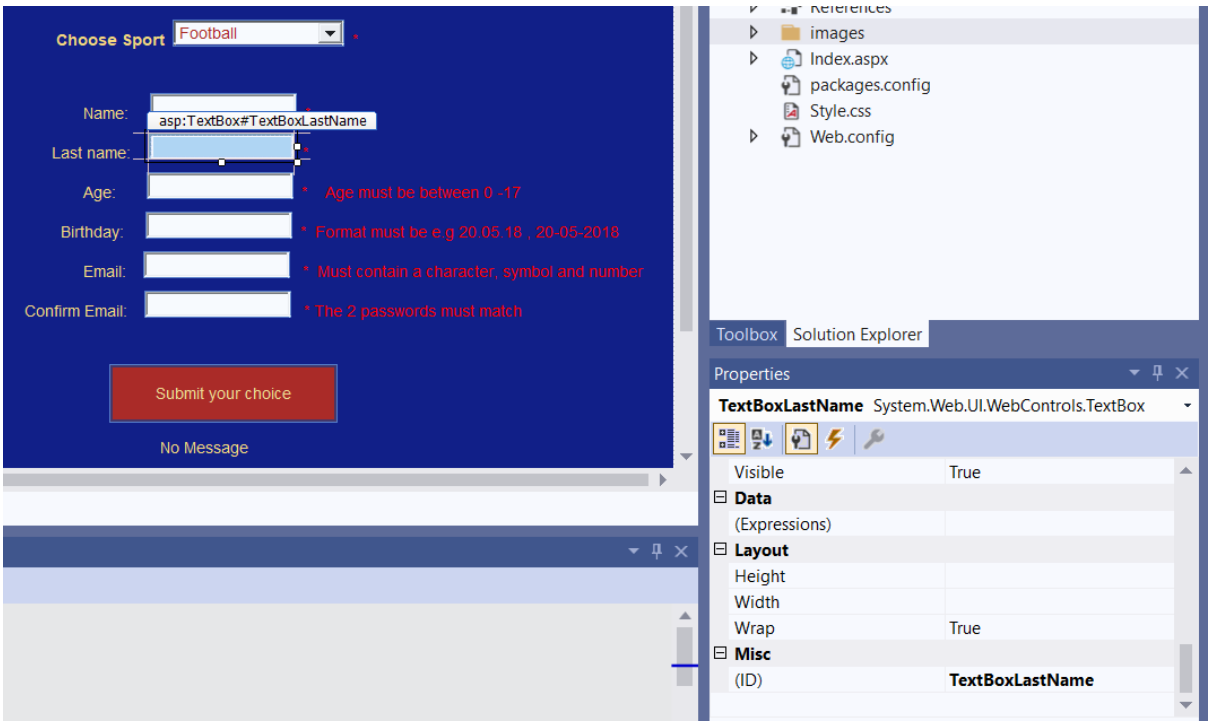
Labels

I put in the data which is described in the handin text, name, last name, age, birthday, email and confirmation email. I have also put in two message labels for a message and special message.



Textboxes

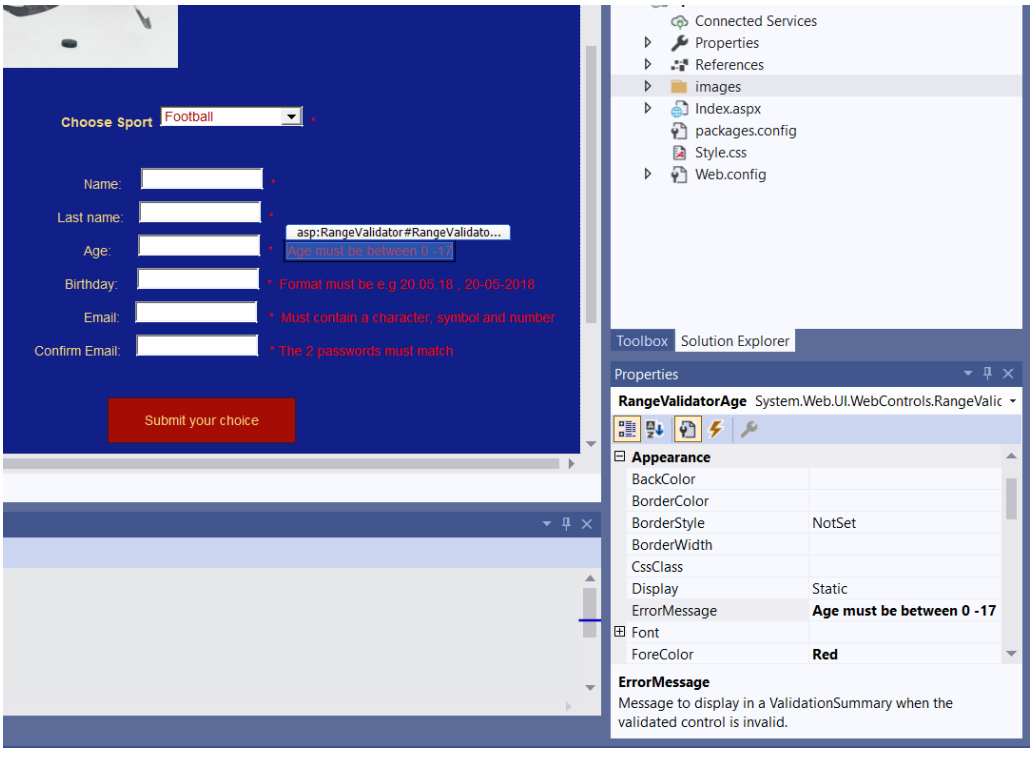
I then created textboxes which correspond to the matching labels so they have their ID name which matches the label name.

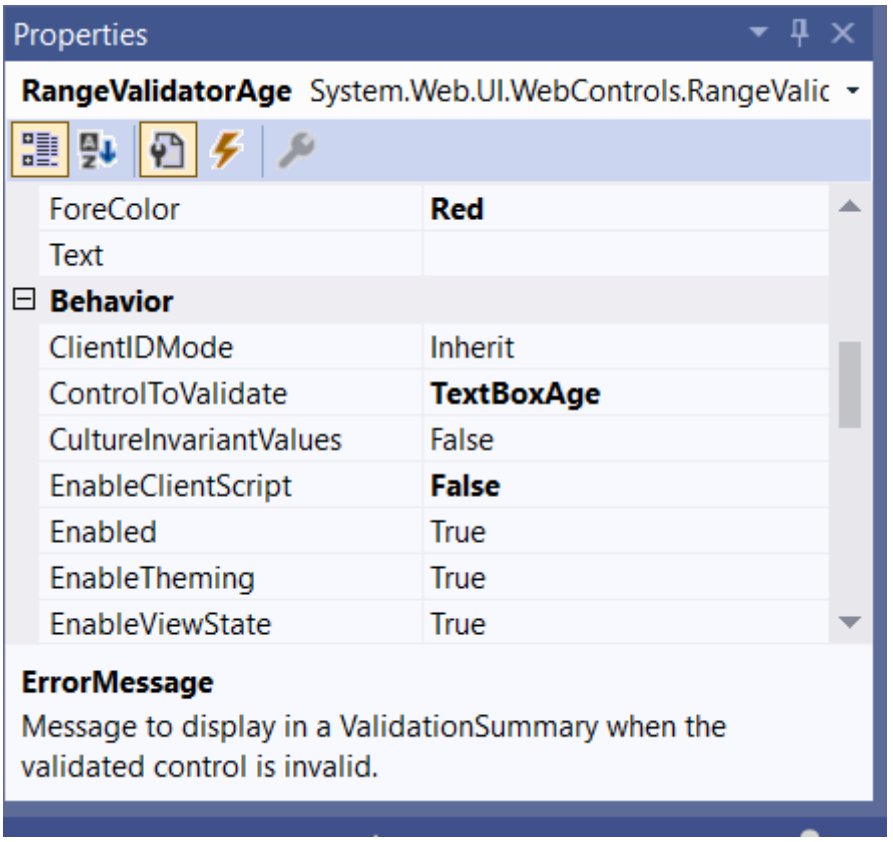


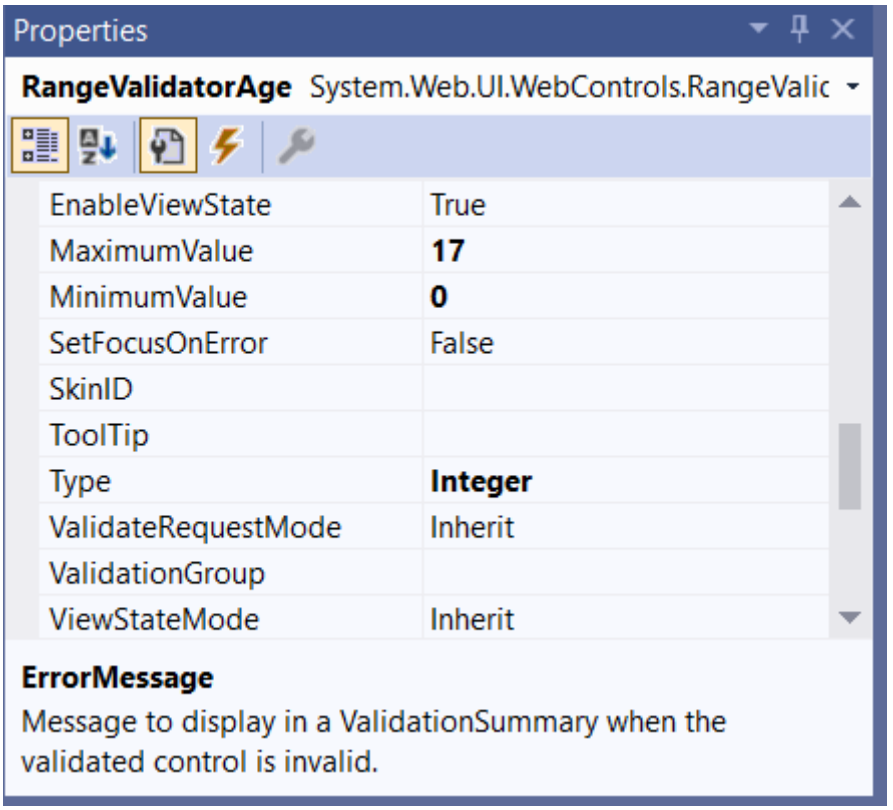
Validators

Age validator

For the age box I had to have the condition of the age limit being between 0-17 therefore I had to use the RangeValidator. I then gave it the ErrorMessage a little sentence explaining the conditions for the box, as well as the color red for error.
Under behavior in properties I connect the matching textbox with the ControlToValidate. In this case that would be the TextBoxAge, so it will check the input made by the user and check if it matches the conditions of the validation.
In order to create the condition age between 0 - 17 I changed the values in the behavior section so MaximumValue = 17 and MinimumValue = 0 and that is how the values are set. I remember to change Type in behavior to integer since it is numbers the user has to input.

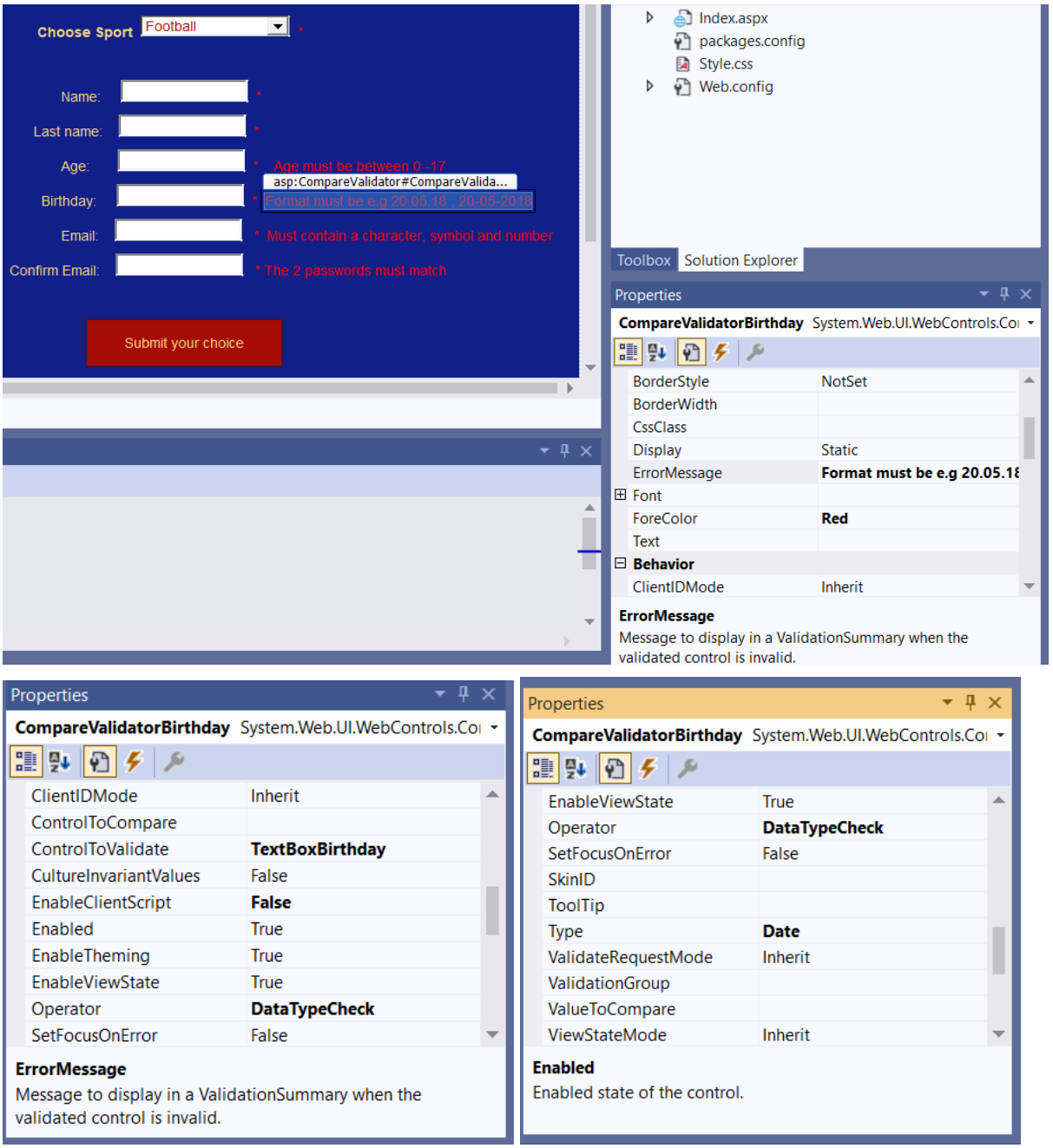



Properties
RangeValidatorAge System.Web.UI.WebControls.RangeValic
ForeColor Red
Text
Behavior
ClientIDMode Inherit
ControlToValidate TextBoxAge
CultureInvariantValues False
EnableClientScript False
Enabled True
EnableTheming True
EnableViewState True
ErrorMessage
Message to display in a ValidationSummary when the validated control is invalid.


Properties
RangeValidatorAge System.Web.UI.WebControls.RangeValic
EnableViewState True
MaximumValue 17
MinimumValue 0
SetFocusOnError False
SkinID
ToolTip
Type Integer
ValidateRequestMode Inherit
ValidationGroup
ViewStateMode Inherit
ErrorMessage
Message to display in a ValidationSummary when the validated control is invalid.

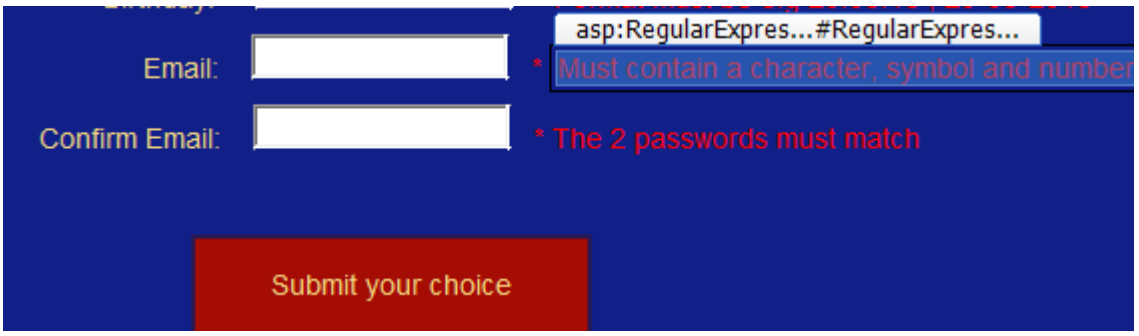
Birthday validator

For the birthday section I use CompareValidator where a lot of the first stuff is the same as the age validator, so the ErrorMessage & Forecolor are the same, but the ControlToValidate is set to match the Birthday textbox. But I changed the Operator to DataTypeCheck which creates a condition that checks if the data type matches the condition we give the validator for the textbox. And then I choose the Type to be Date so the program knows that it has to check that the user puts in the correct data type, that being a date setup.

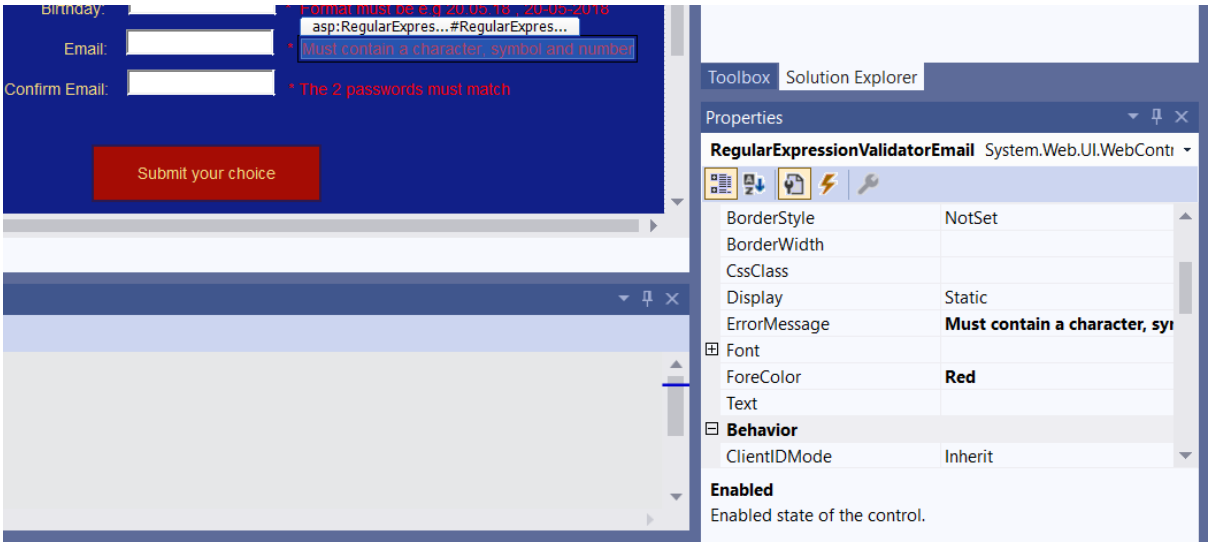


Email Validator

For the Email section I needed to have two boxes, one for the email and another to compare and confirm the email.



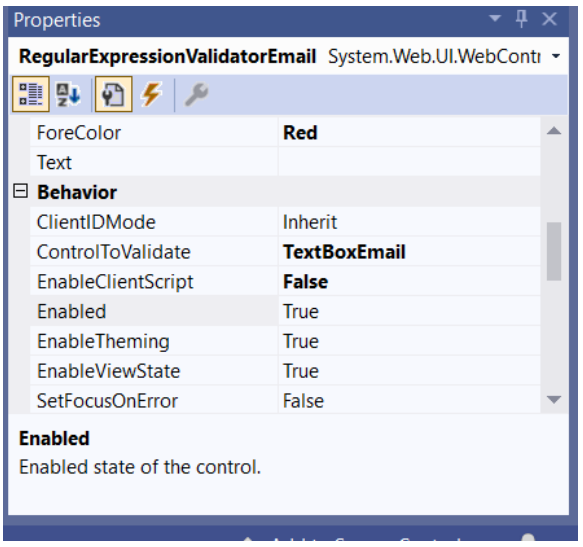
The Email box I use RegularExpressionValidator and once again a lot of the same information are used the same way as the earlier ones.

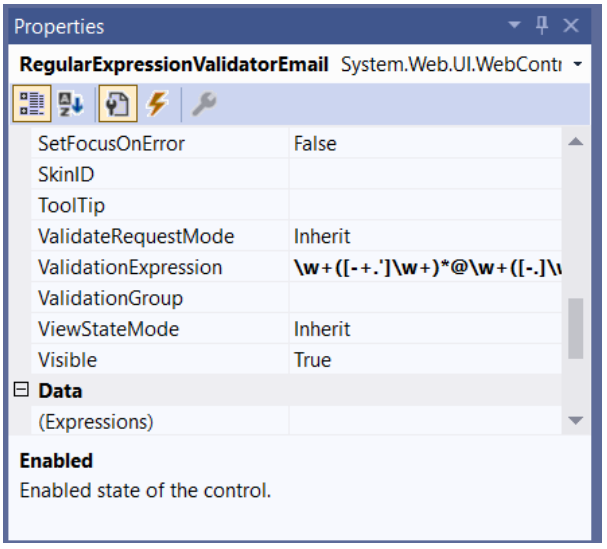


After connecting the TextBoxEmail to the RegularExpressionValidator I then under ValidatorExpression chose the Internet e-mail address which is the built-in mail expression.

```
\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*
```

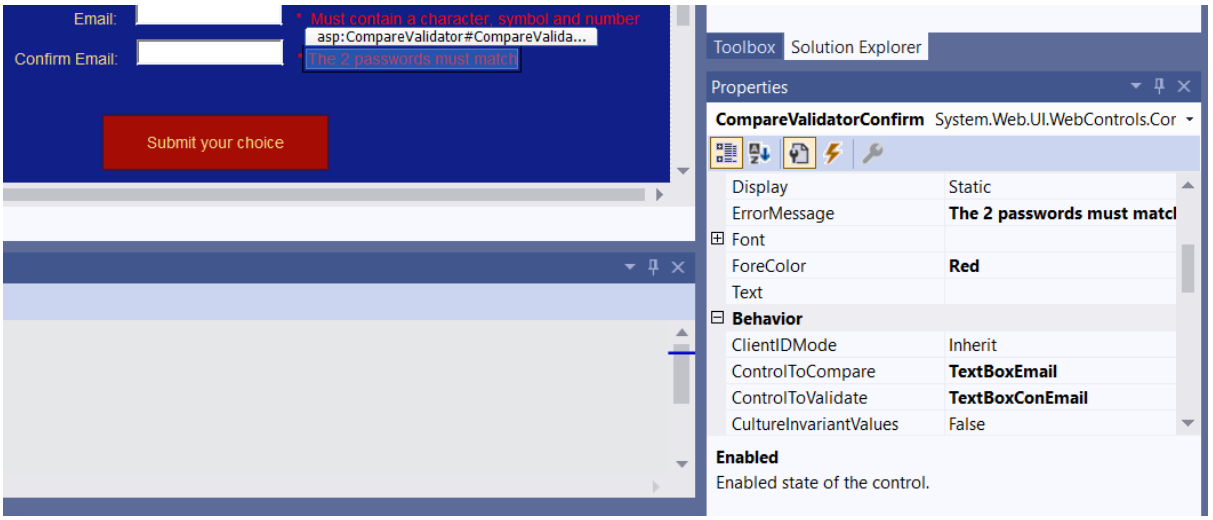
Which should allow the user to write a mail that has to contain a @ and .(com/dk ect…).





Confirm Email Validator

The confirm validation I used CompareValidator again a few things are done in the same way as the other validators.
But to be able to tell the program that the TextboxConEmail is the exact same as the email text box, I chose ControlToCompare to the TextBoxEmail, and ControlToValidate to the TextboxConEmail. This way it knows that is has to compare with the email section and has to be the same, and the box it has to validate is the confirm email textbox.



Button

I put in one button to submit all the data, and connected it to the message labels as talked about earlier. They are gonna write a message depending on the outcome of the data input. Using if else statements I tell the program how to connect the labels to the button and what to write depending on the outcome.

```
protected void ButtonSubmit_Click(object sender, EventArgs e)
        {
            if (Page.IsValid)
            {
                LabelMessage.Text = "Valid page";
                LabelSpecialMessage.Visible = false;
            }
            else
            {
                LabelMessage.Text = "Error on page. Data will not be
used";
                LabelSpecialMessage.Text = "Correct the errors";
                LabelSpecialMessage.Visible = true;
            }
        }
```

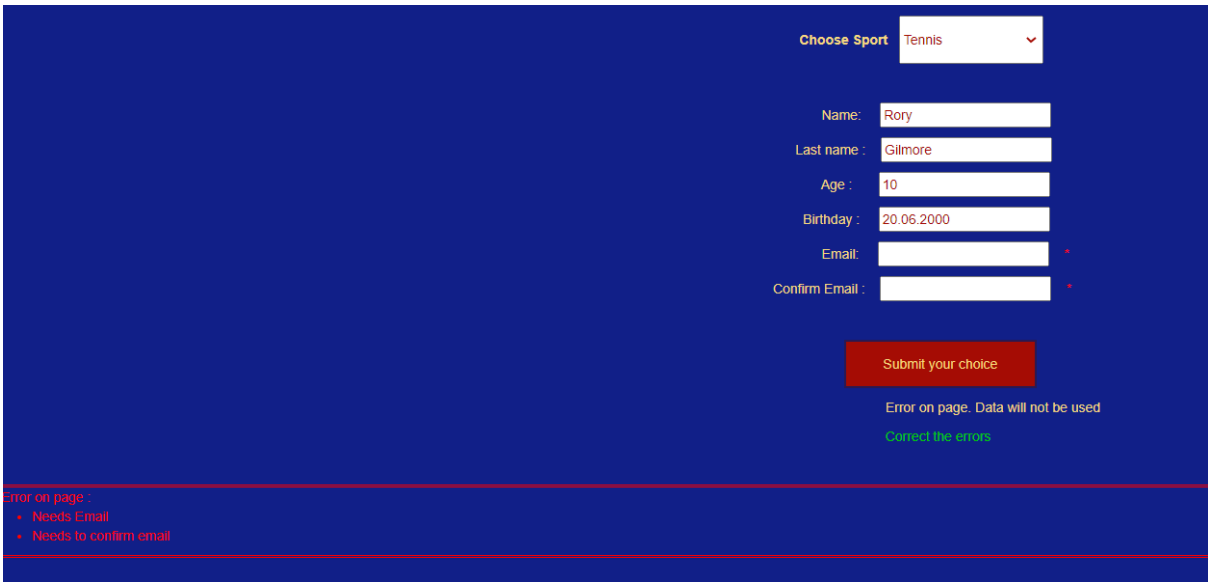
It tells the program if all the data put in are valid, therefore making the page valid then it should write ‘’valid page’’, and ‘’correct the errors’’ if there are errors.

Required Fields

On each of the fields I put in the RequiredFieldValidator where I set up the requiredfieldvalidator (ControlToValidate) to their respective textboxes. This way we tell the program that all the fields are required to be filled in before pressing the submit button otherwise there will be an error.

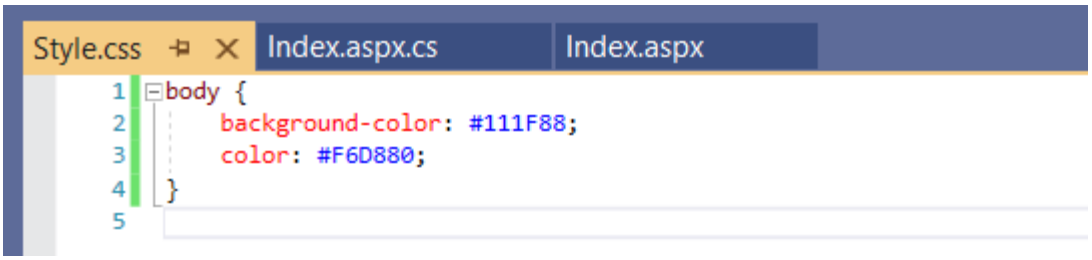
Validation Summary

The ValidationSummary will show the user the errors that may occur and need to be fixed as well as the error messages created earlier.



Style

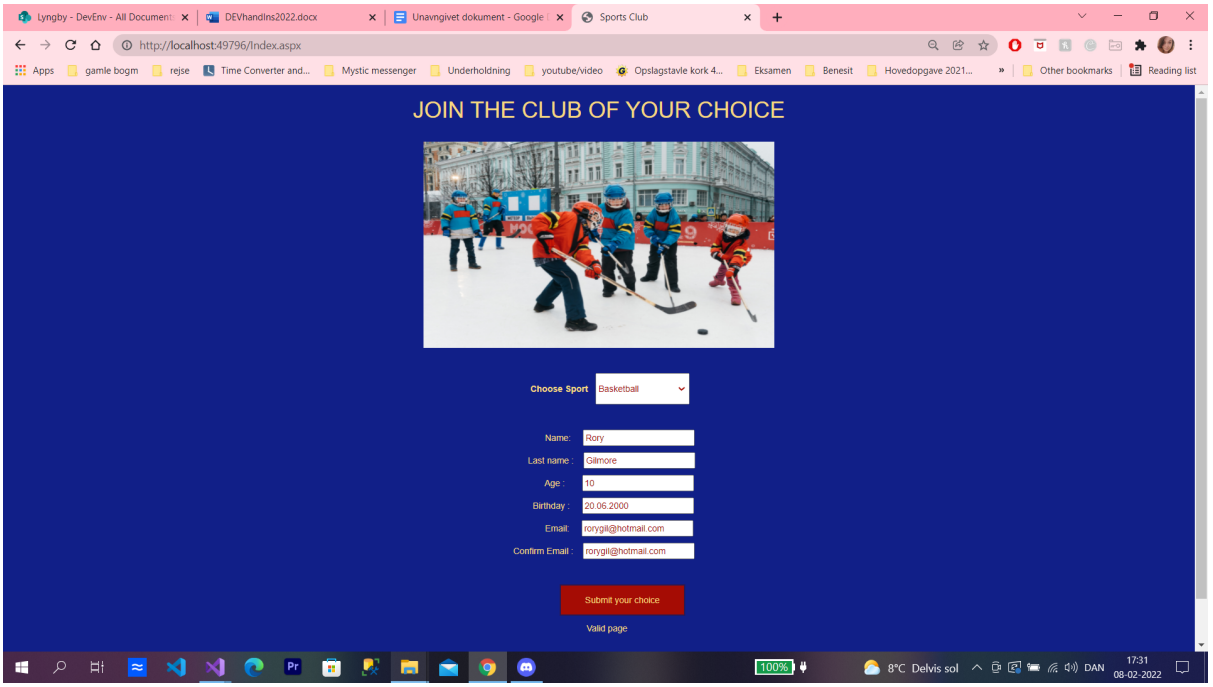
I also used a stylesheet to very simply change the background and text color.



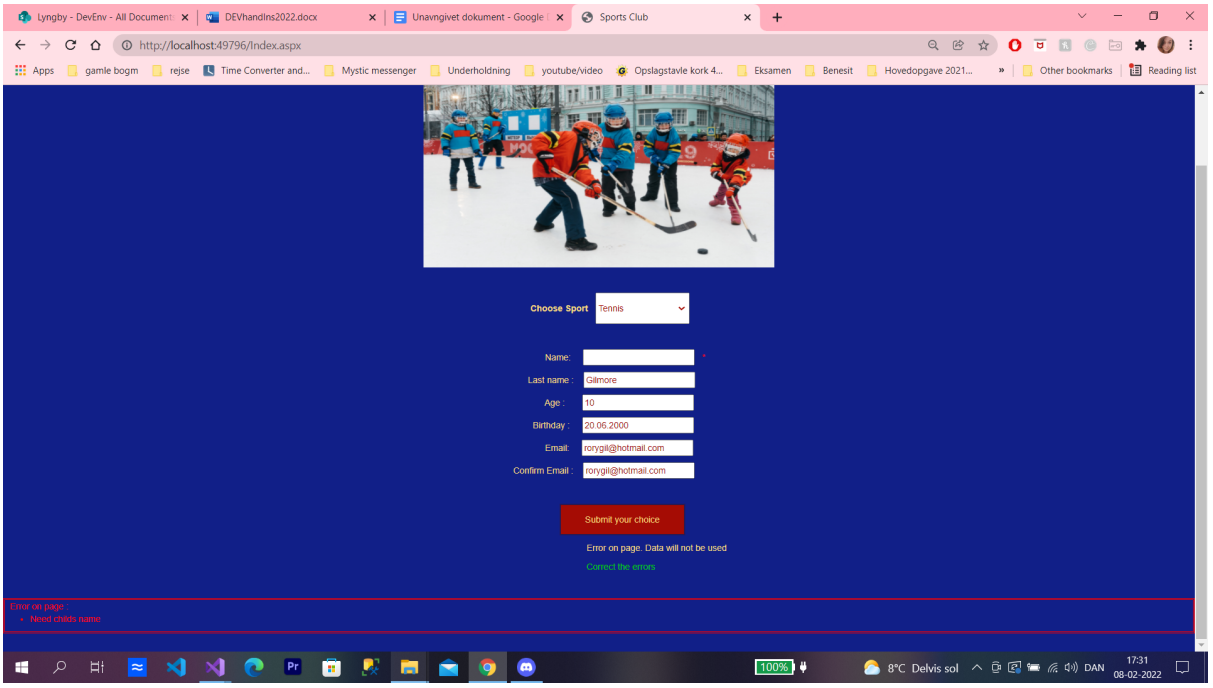
Page test

I will show you how the page looks when tested to see how it looks when inputting both the correct or wrong data.

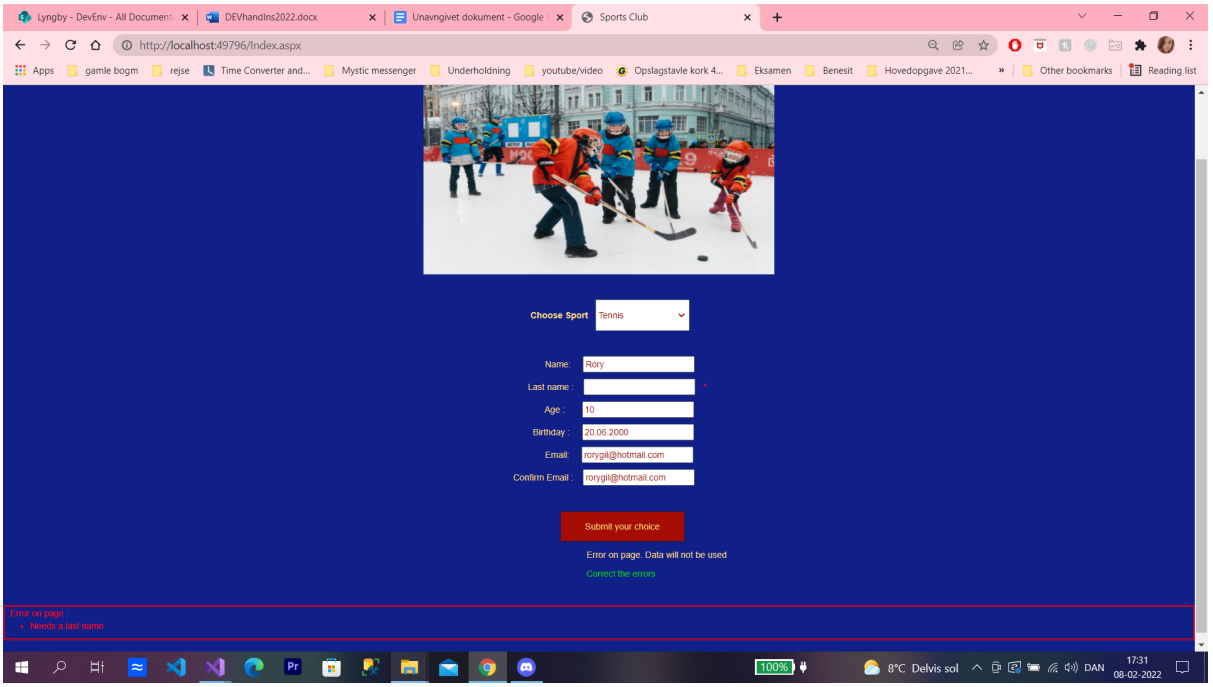
Valid page:



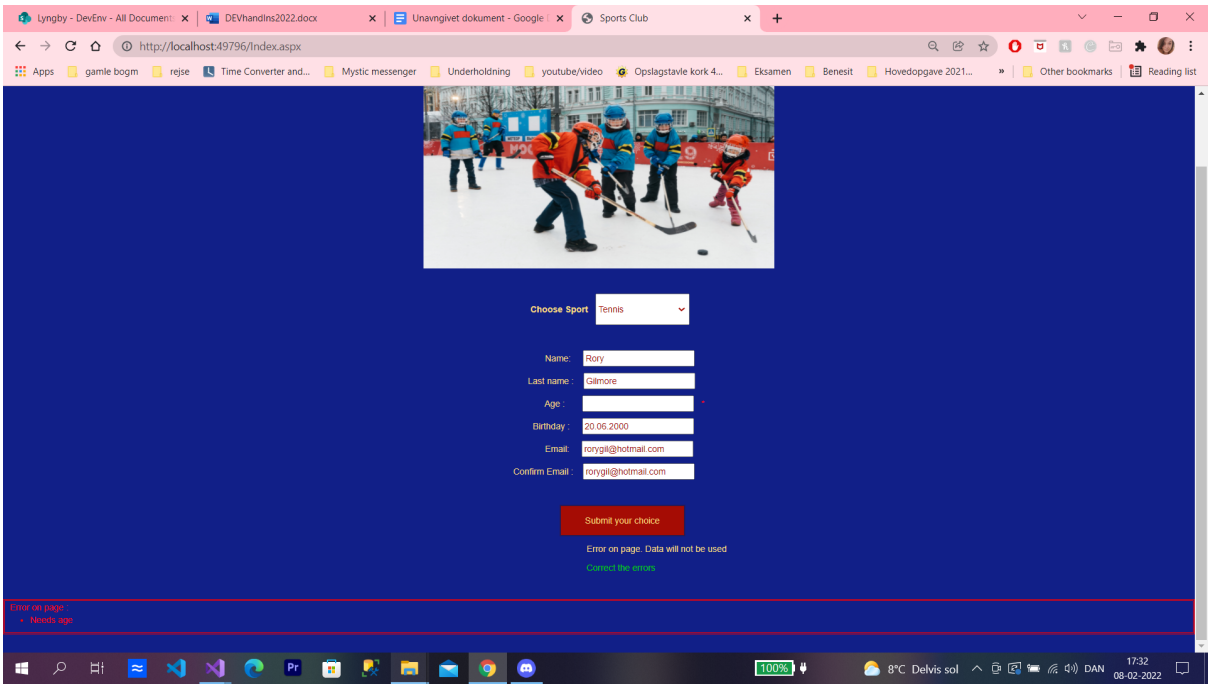
No name:



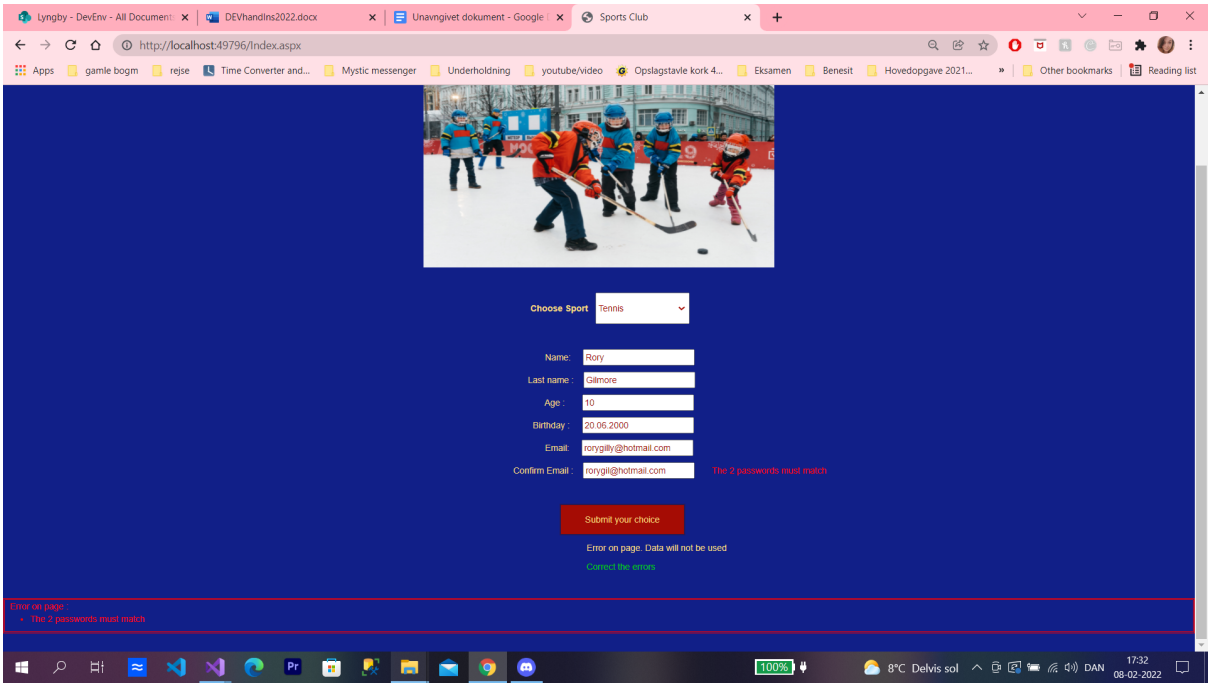
No last name:



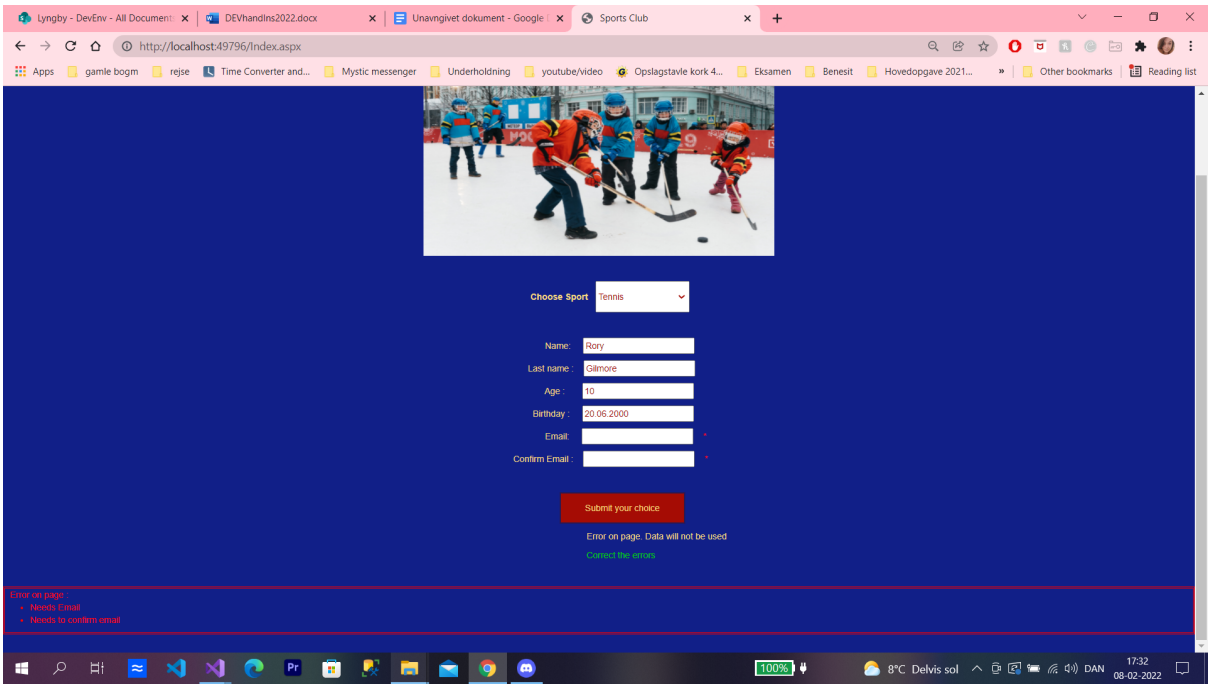
Age mistake:

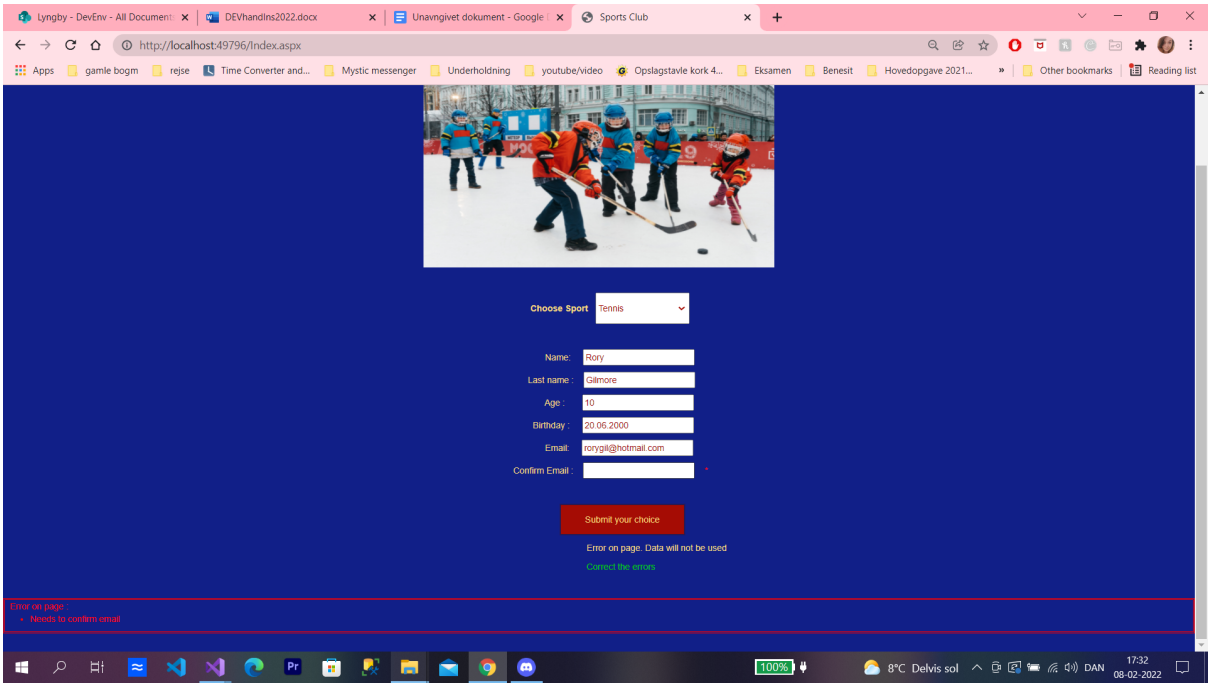


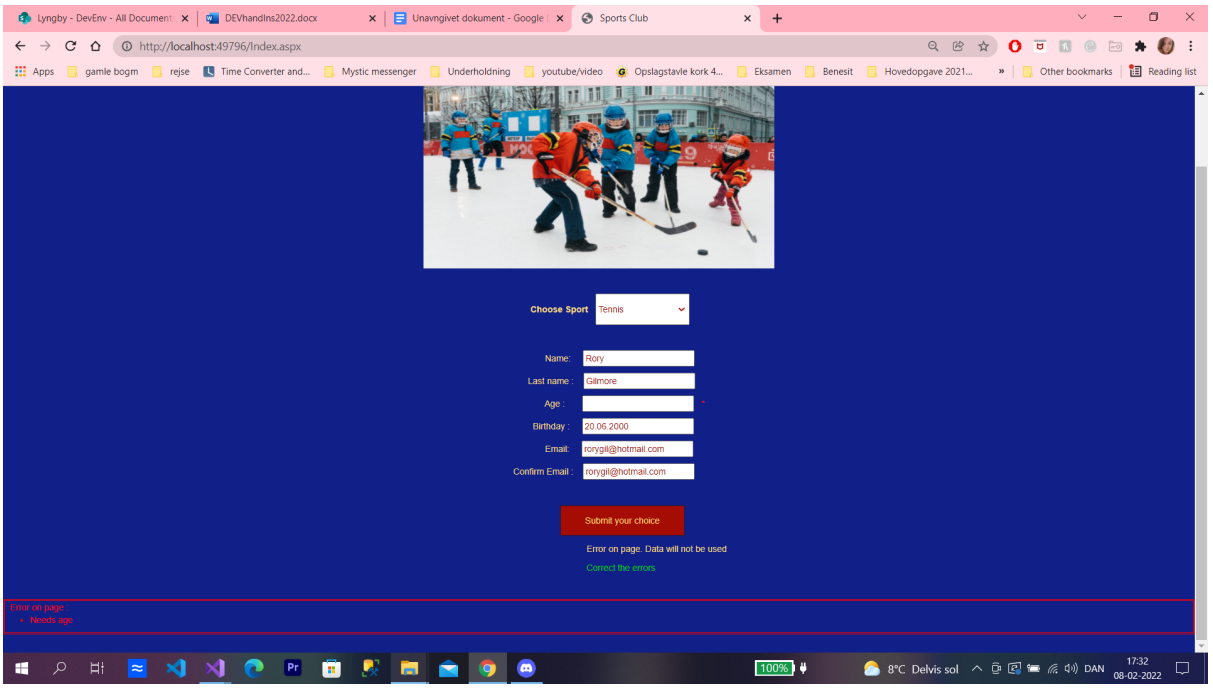
The image here shows what happens if you input an age value that are higher than the maximum value we chose (17):



Email errors:




Lyngby - DevEnv - All Document
DEVhandIns2022.docx
Unavngivet dokument - Google
Sports Club
http://localhost:49796/Index.aspx
Choose Sport
Tennis
Name:
Rory
Last name :
Gilmore
Age :
10
Birthday :
20.06.2000
Email:
rorygil@hotmail.com
Confirm Email :
Submit your choice
Error on page. Data will not be used
Correct the errors
Error on page :
Needs to confirm email


Lyngby - DevEnv - All Document
DEVhandIns2022.docx
Unavngivet dokument - Google
Sports Club
http://localhost:49796/Index.aspx
Choose Sport
Tennis
Name:
Rory
Last name :
Gilmore
Age :
Birthday :
20.06.2000
Email:
rorygil@hotmail.com
Confirm Email :
rorygil@hotmail.com
Submit your choice
Error on page. Data will not be used
Correct the errors
Error on page :
Needs age