

**Activity 2**

Jason Hatfield

CST-350

7/4/2023

Professor Bradley Mauger

## Part 1 Controllers and Views

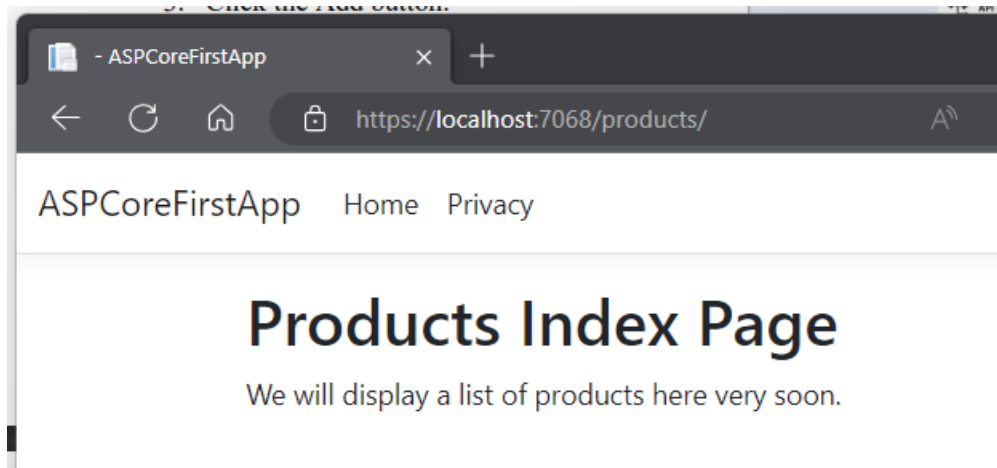


Figure 1 Products Index View Page

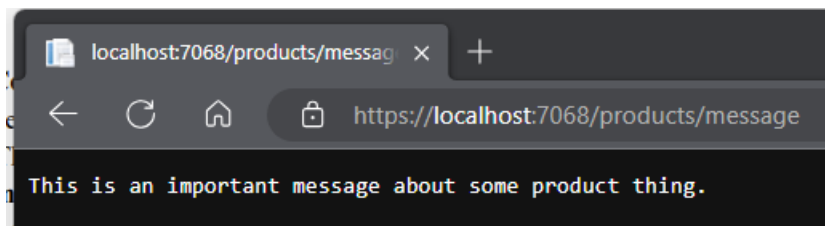


Figure 2 Products Index Message Controller Page

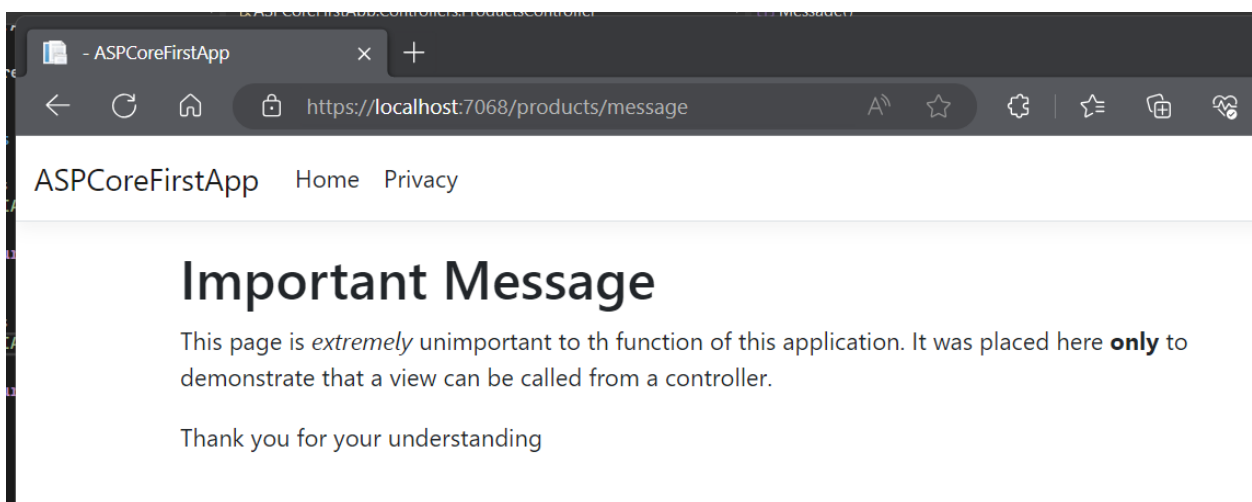
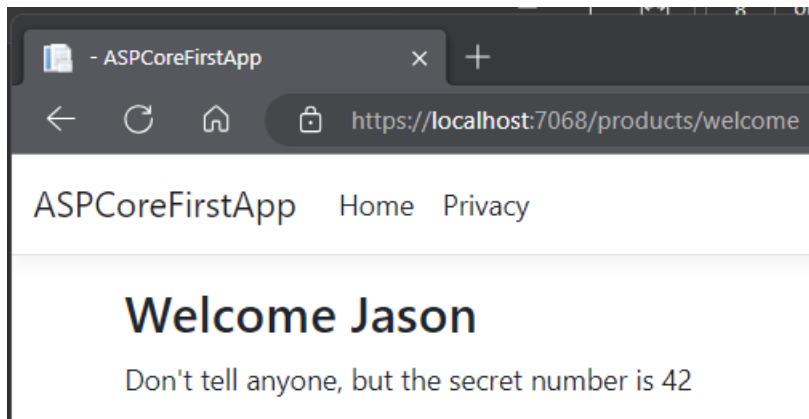


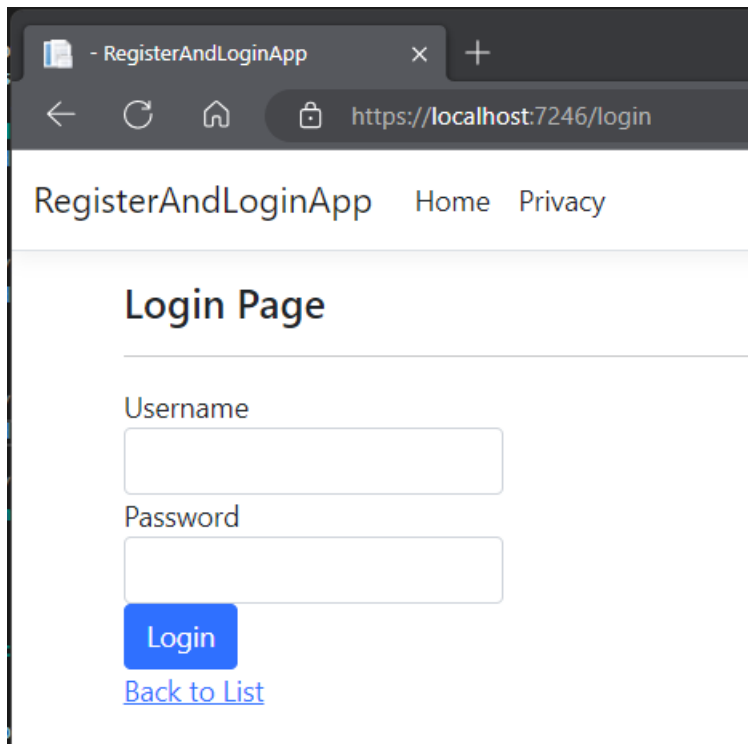
Figure 3 Message from the products Message View page.



*Figure 4 ViewBag.name and ViewBag.secretNumber*

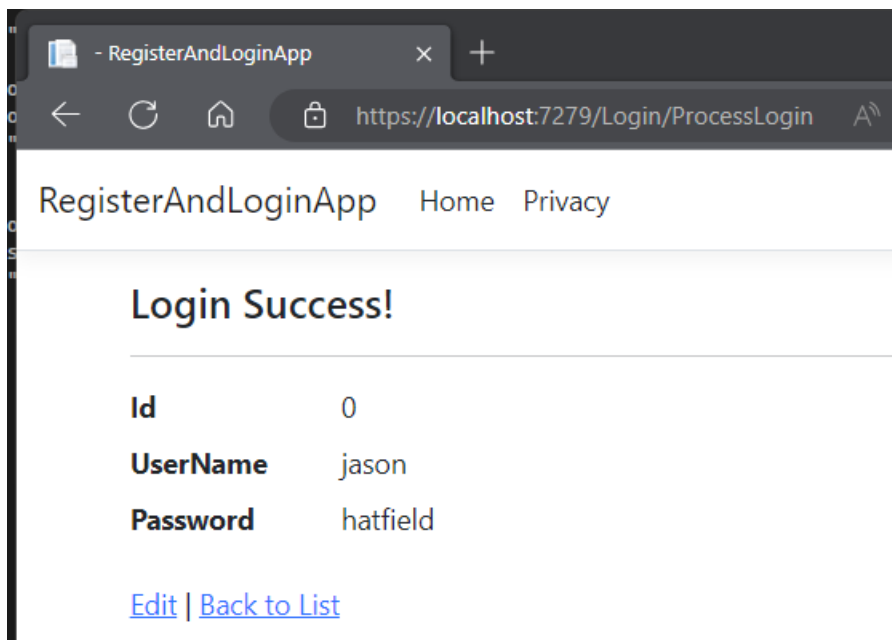
This activity explored the fundamentals of Controllers and Views in ASP.NET Core. I created a 'ProductsController' and associated Views, learning to pass parameters via the URL and display them in the browser. I also learned about routing, IActionResult, and the use of ViewBag for data transfer. This exercise provided a hands-on understanding of the MVC (Model-View-Controller) pattern in ASP.NET.

## Part 2 Login



The screenshot shows a web browser window with the title "RegisterAndLoginApp". The address bar displays "https://localhost:7246/login". The page has a navigation bar with "RegisterAndLoginApp", "Home", and "Privacy". The main content area is titled "Login Page" and contains a form with two input fields: "Username" and "Password". Below the "Password" field is a blue "Login" button and a blue link labeled "Back to List".

Figure 5 Login Page



The screenshot shows a web browser window with the title "RegisterAndLoginApp". The address bar displays "https://localhost:7279/Login/ProcessLogin". The page has a navigation bar with "RegisterAndLoginApp", "Home", and "Privacy". The main content area is titled "Login Success!". Below the title is a table displaying the login details:

<b>Id</b>	0
<b>UserName</b>	jason
<b>Password</b>	hatfield

Below the table are two blue links: "Edit" and "Back to List".

Figure 6 Successful Login

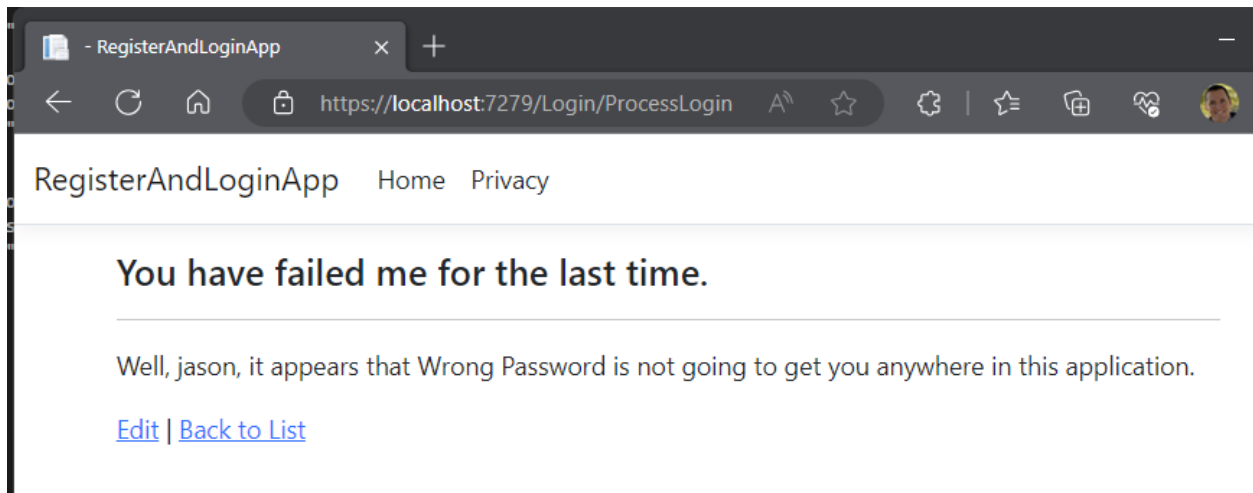


Figure 7 Failed Login

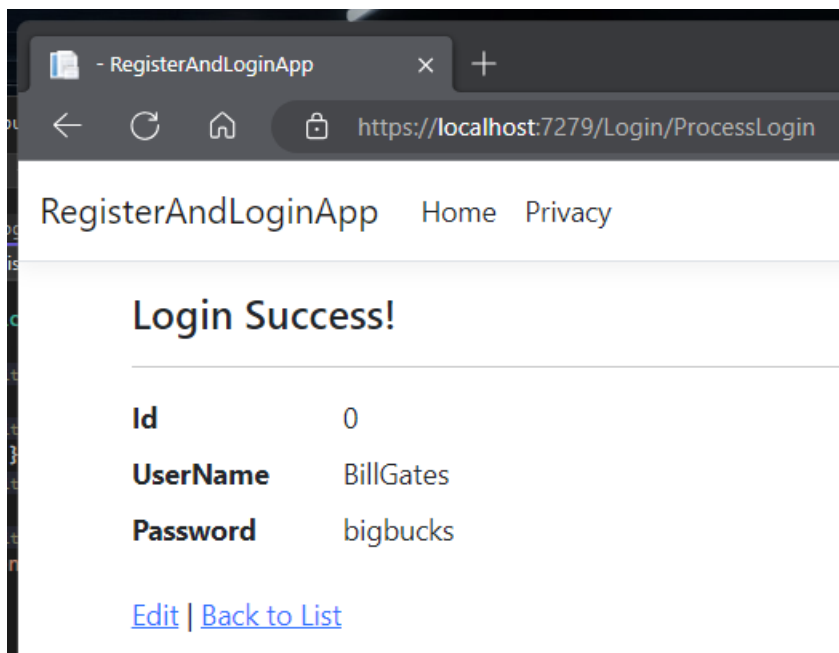


Figure 8 Successful login using modified Login controller.

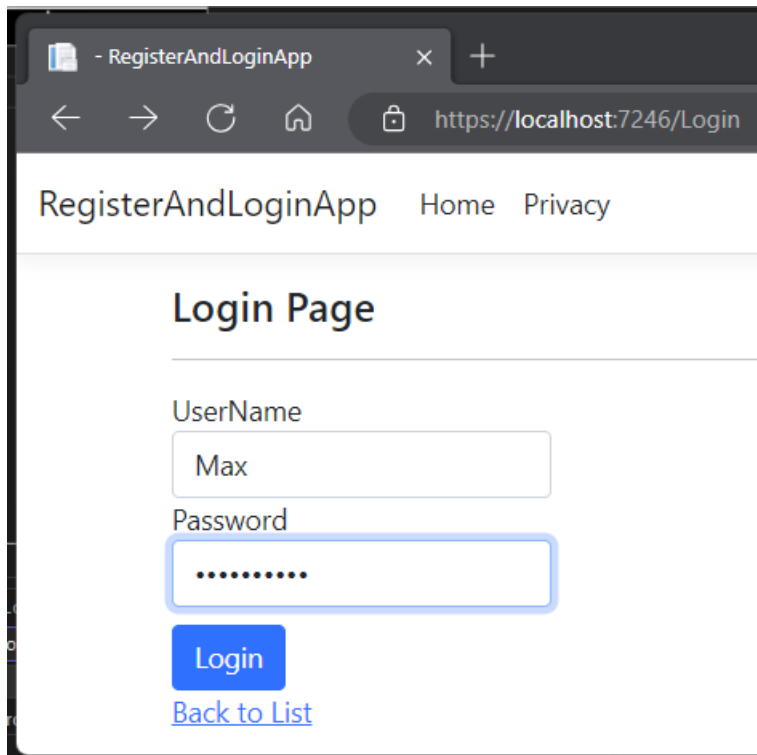


Figure 9 Login Page

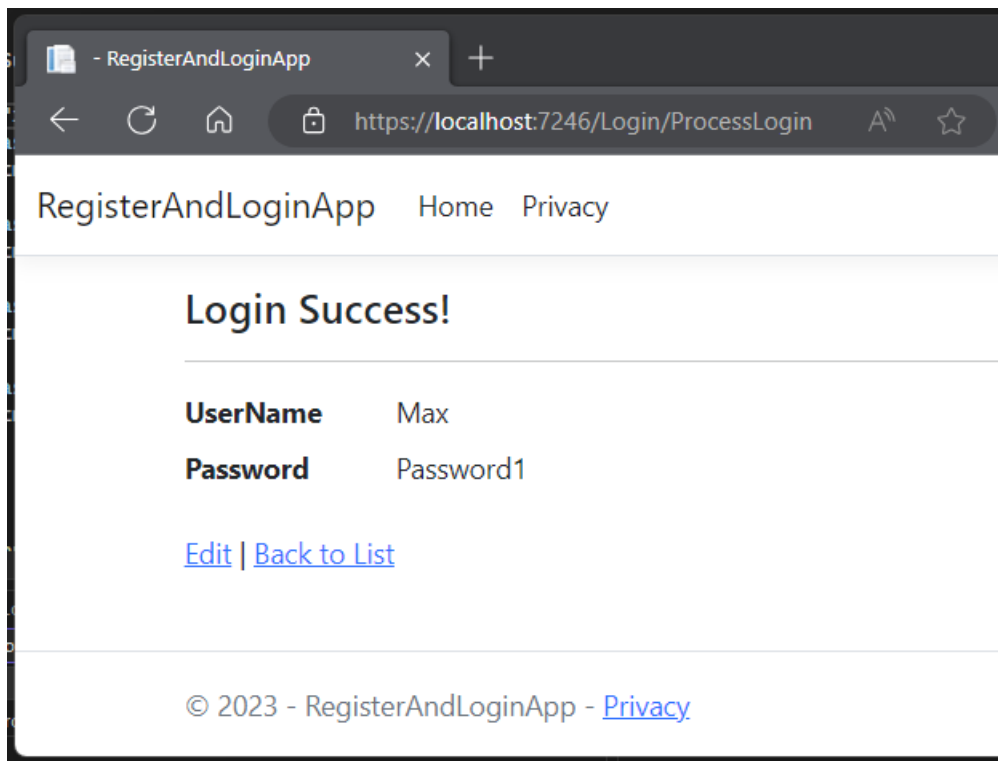


Figure 10 Login Success Using DB Credentials

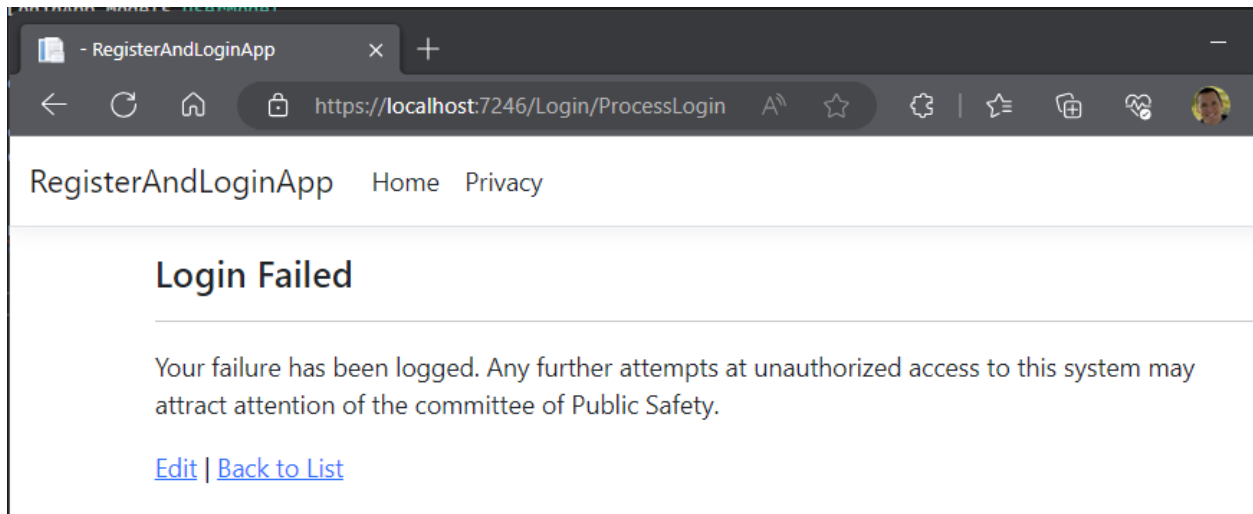


Figure 11 Login Failure using DB Credentials

This lesson demonstrates key concepts such as MVC Architecture, Controllers and Views, Models, Data Security, Databases, and Services in N-Layer Design. It walked me through creating a login system in a .NET MVC application, including steps to create a login view, route, controller, user model class, and data security class. The lesson also covers binding the model to the view, creating a database schema, authenticating users using a database, and implementing a Data Access Object (DAO) for handling SQL queries. It underscores the importance of a clear and structured flow of operations in a layered design and using SQL commands to interact with the database.

## Part 3 Data Validation

The screenshot shows a web browser window with the title '- AppointmentMaker'. The address bar displays 'https://localhost:7203/appointments'. The page has a navigation bar with 'AppointmentMaker', 'Home', and 'Privacy' links. The main heading is 'AppointmentModel'. Below it, there are several form fields with validation errors:

- Patient's Full Name**: The input field contains 'me'. Below it, a red error message states: 'The field Patient's Full Name must be a string with a minimum length of 4 and a maximum length of 20.'
- Appointment Request Date**: The input field contains '07/17/2020' and has a calendar icon on the right.
- Patient's Net Worth**: The input field is empty.
- Primary Doctor's Last Name**: The input field is empty.
- Patient's perceived level of pain (1 low to 10 high)**: The input field contains '22'. Below it, a red error message states: 'The field Patient's perceived level of pain (1 low to 10 high) must be between 1 and 10.'

At the bottom of the form, there is a blue 'Create' button and a blue link labeled 'Back to List'.

Figure 12 Appointment Maker with imposed requirements



## AppointmentModel

---

Patient's Full Name

Jason Hatfield

Appointment Request Date

07/05/2025



Patient's Net Worth

25000

Doctors refuse to see patients unless their net worth is more than \$90,000

Primary Doctor's Last Name

Smith

Patient's perceived level of pain (1 low to 10 high)

6

Doctors refuse to see patients unless their pain level is above a 5

Create

[Back to List](#)

Figure 13 Programming Challenge

## Part 4 Button Grid – Manage Dynamic Objects on a View

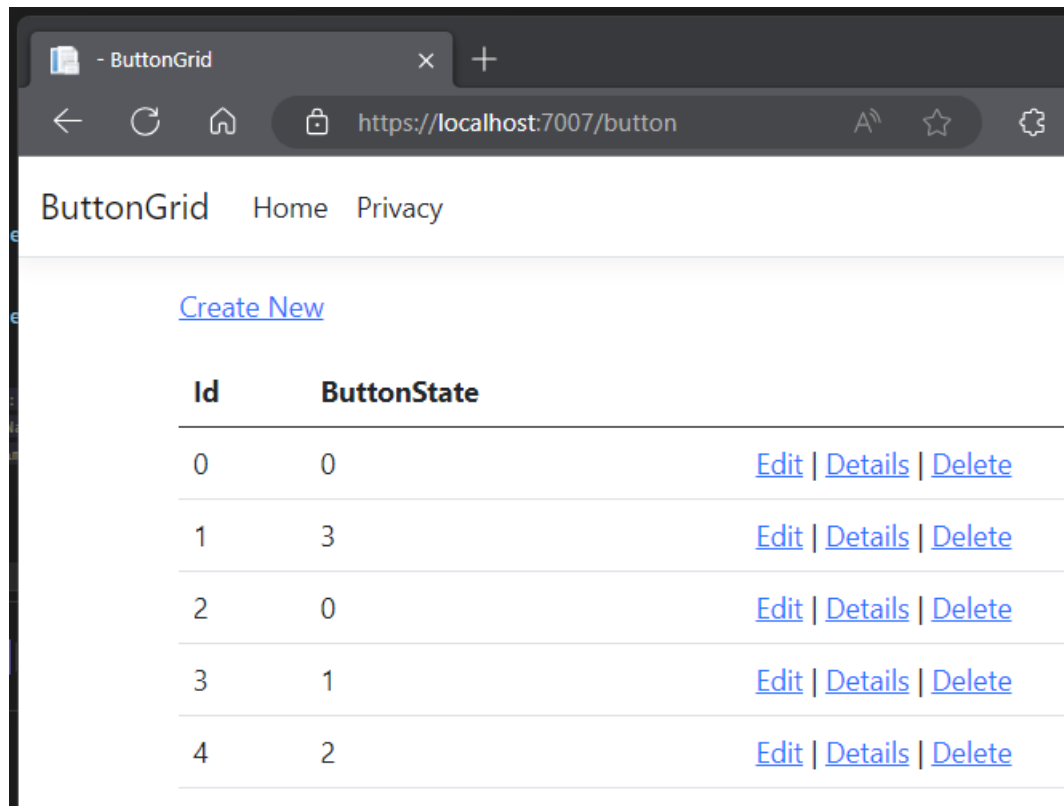


Figure 14 Button State

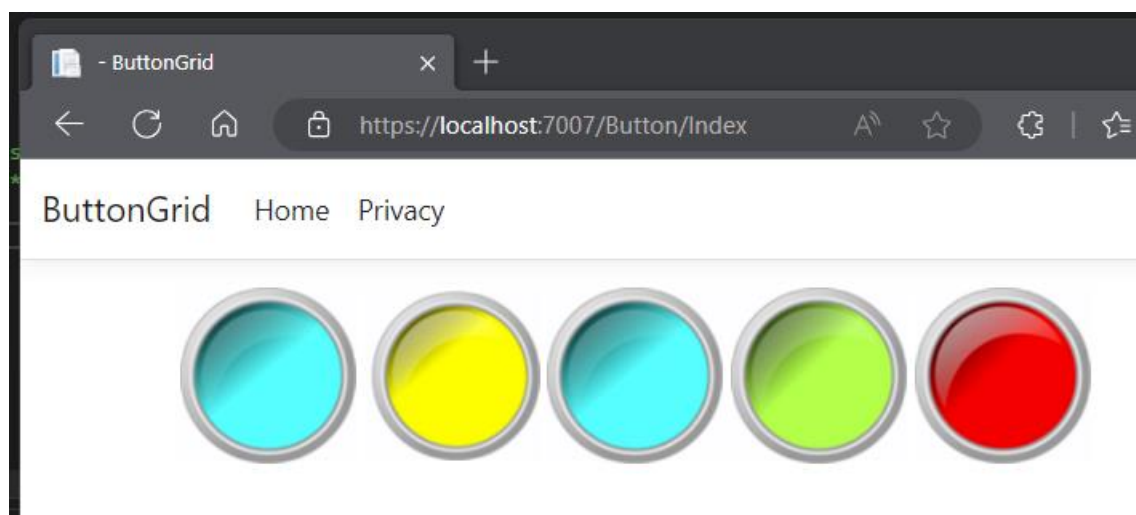


Figure 15 Buttons!

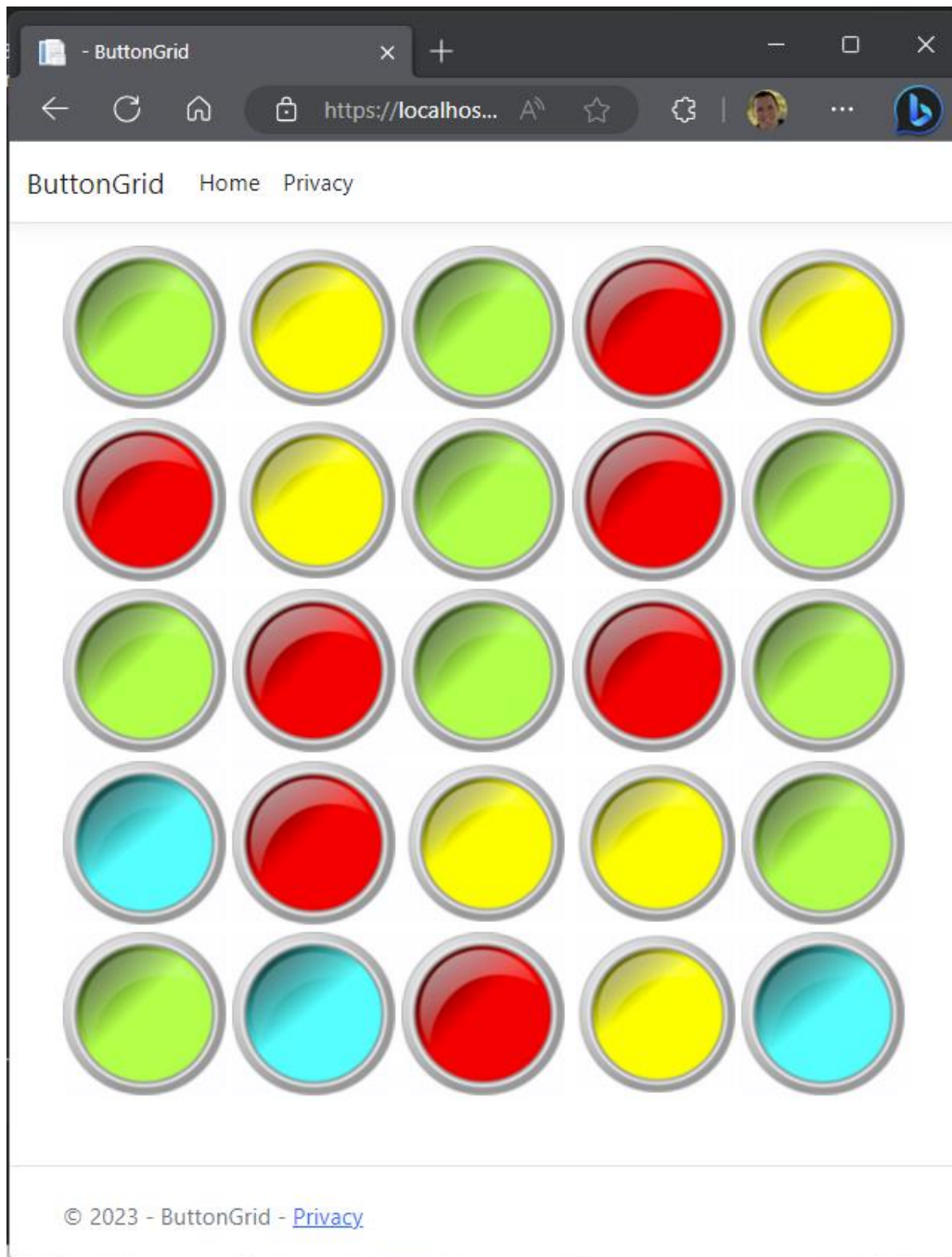


Figure 16 25 Random Buttons!

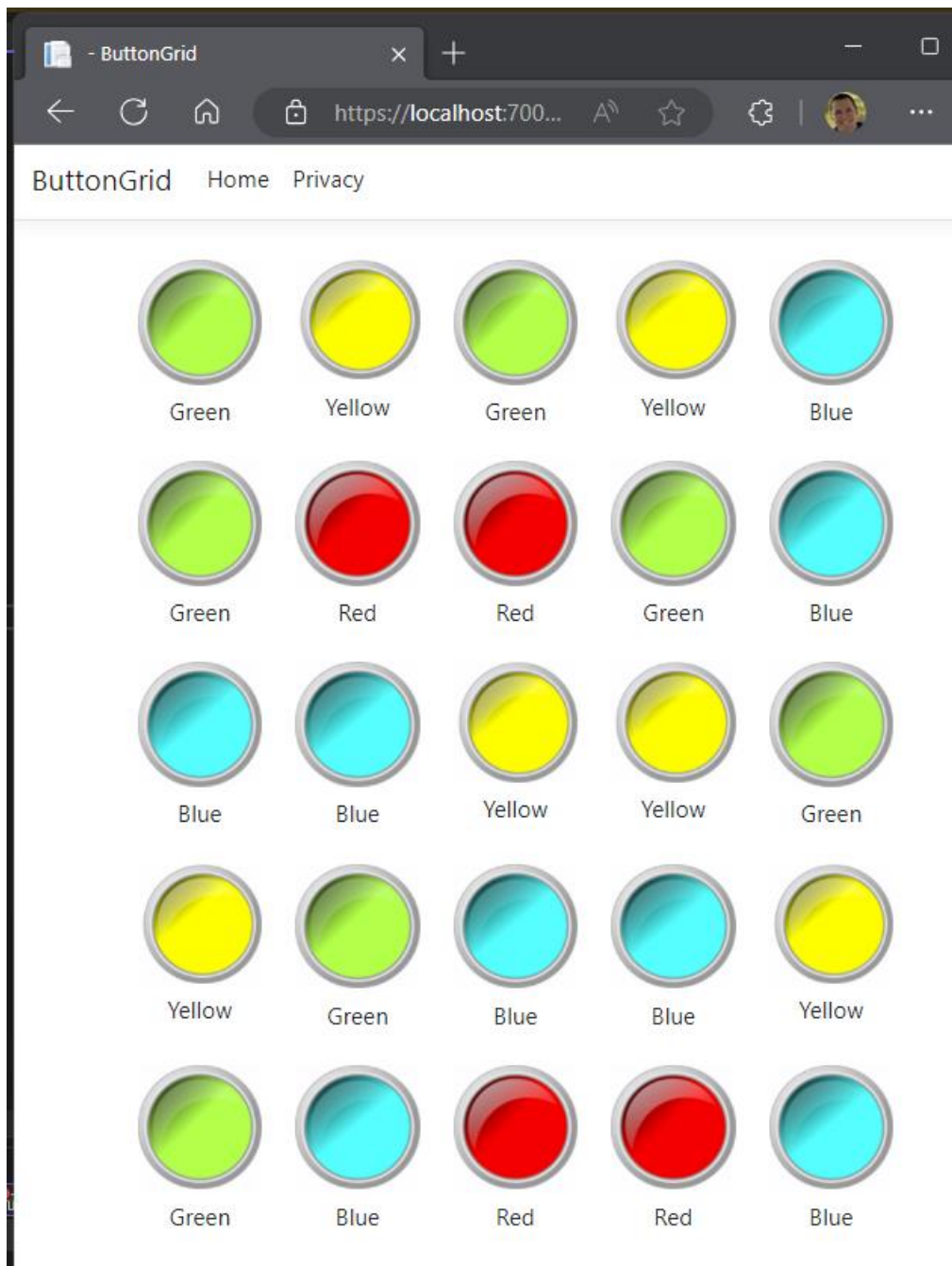


Figure 17 Buttons using FlexBox

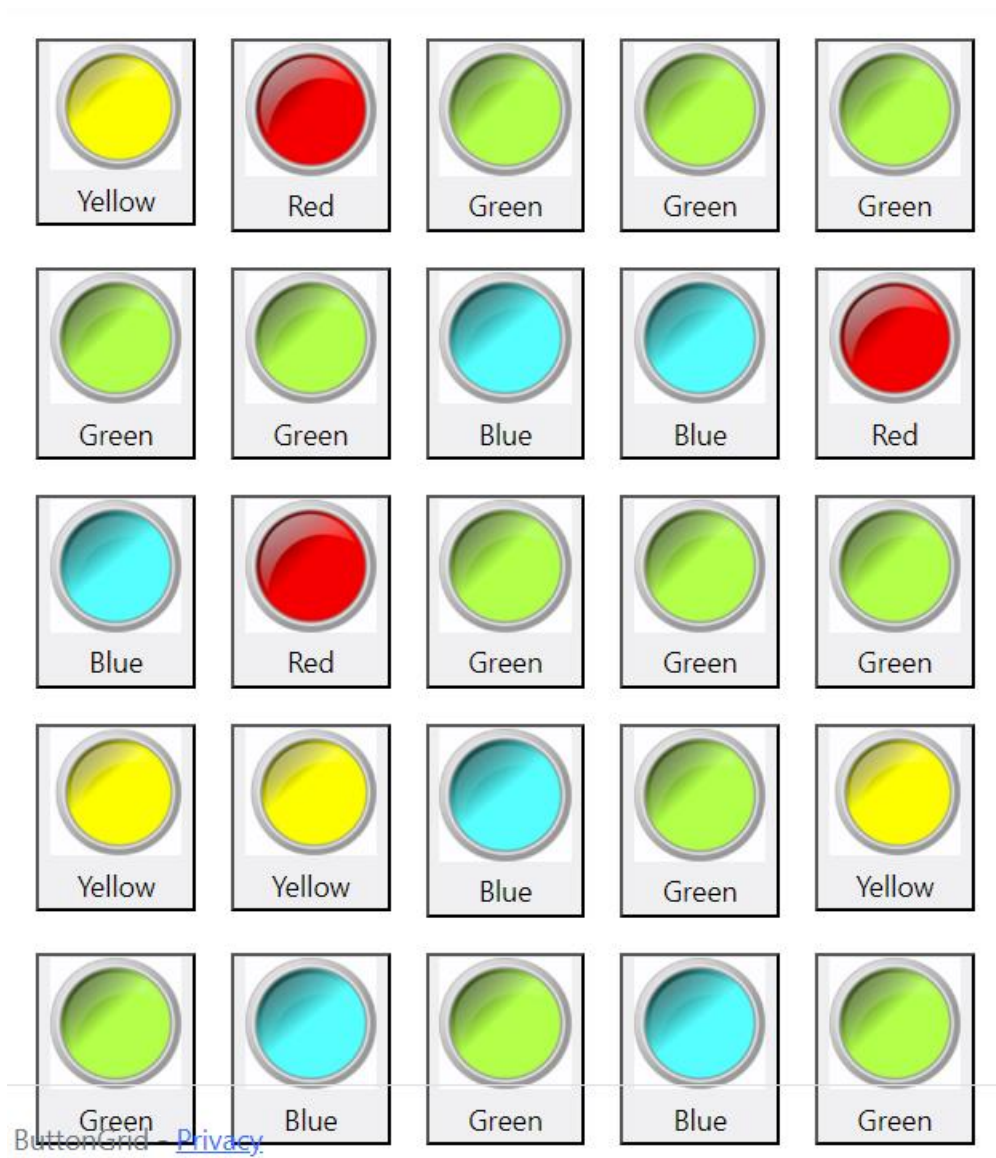
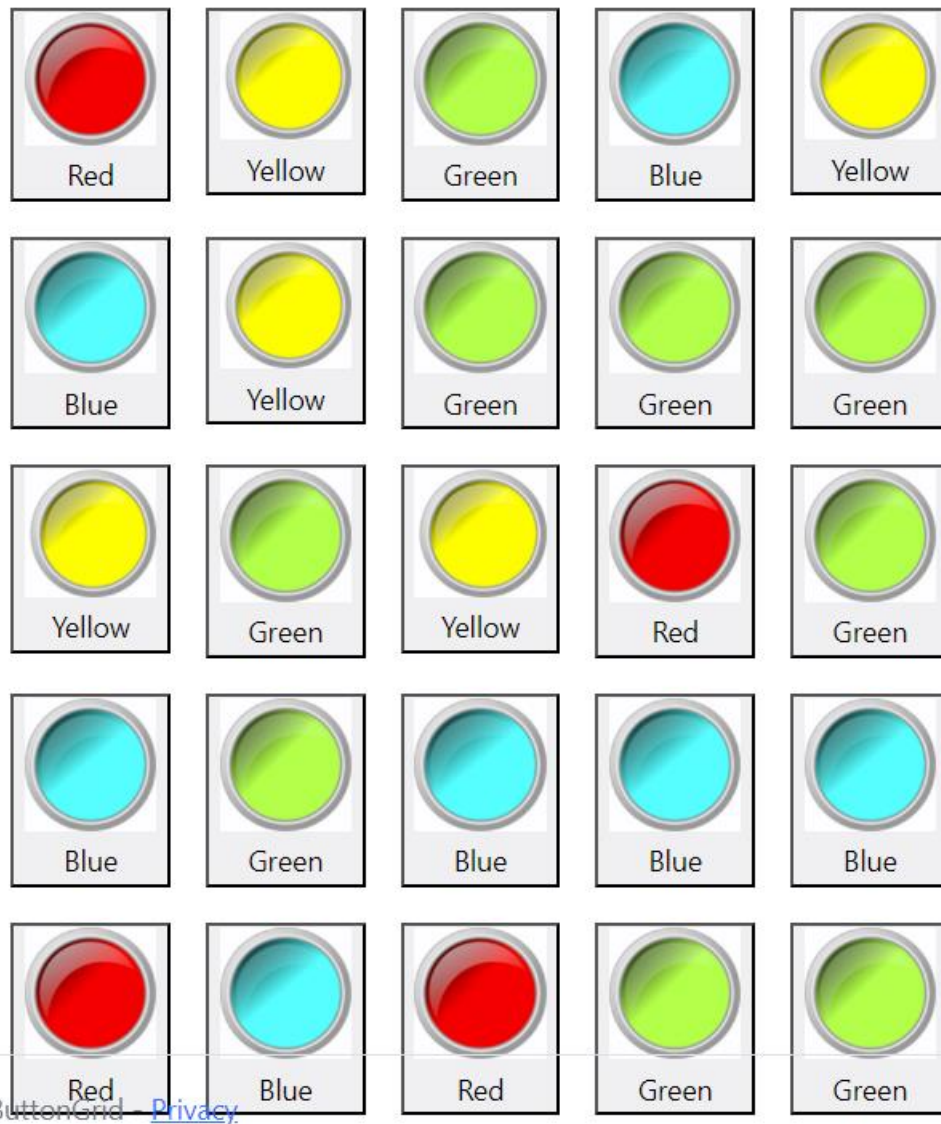


Figure 18 State 1



3uttonGrid - [Privacy](#)

Figure 19 State 2