

BeMyEye Exercise

Angular numerical password Input

Objective

Using **AngularJS** or **Angular X** you have to create a **new** Numerical password Input **component** following the given **technical specification**.

Mockup

A mockup of a numerical password input component. It consists of a rectangular container with a black border. Inside the container, at the top, is a horizontal rectangular display field with a black border, containing the text '045107764'. Below the display field is a 3x4 grid of square buttons, each with a black border. The buttons contain the following numbers from left to right, top to bottom: Row 1: 6, an empty box, 9, 2; Row 2: 7, 8, 3, 5; Row 3: 1, 0, an empty box, 4.

Feature

This component needs to:

- Collect the user numerical password
- Display the user input

- Prevent the Bots from using the system by randomizing the numerical buttons order
- Compare the user inputs with the password value and tell the rest of the app if the values match

Specification

You will create **1 component** and **2 sub-components**:

- NumericalPasswordInput
 - PasswordViewer
 - RandomizedNumericInput

NumericalPasswordInput

Inputs

userPassword: **String**

Outputs

passwordMatch: **Void**

Role

This component will:

- Be the bridge between the **PasswordViewer** and the **RandomizedNumericInput**
- When the user has entered **9 digits**, compare the user input with the **userPassword**. If the values match, **trigger** the passwordMatch event else **reset** the current password

PasswordViewer

Inputs

currentPassword: **String**

Role

This component will display the **currentPassword** value

RandomizedNumericInput

Inputs

currentPassword: **String**

Outputs

currentPasswordChange: **EventEmitter<String>**

Role

This component will:

- Display a grid of 12 buttons, 1 for each numerical value and 2 blanks.
- Randomize the order of the buttons
- Trigger **currentPasswordChange** passing **currentPassword** every time the user enters a new value