

# Frontend Coding Challenge

## Task: *Registration form*

Please implement a registration form component that consists of two inputs (E-Mail & Password) with labels, a password strength indicator and a submit button.

Enable the submit only for passwords that satisfy all of the criteria below:

- *8 or more characters*
- *1 or more lowercase letters*
- *1 or more uppercase letters*
- *1 or more decimal characters*
- *1 or more special characters*

The image displays three different design mockups for a registration form, each titled 'Registration' in a black header bar.   
1. The first mockup (left) features a white background with a central form box. Inside, there are two input fields: 'Email' (containing 'test@test.de') and 'Password' (containing '\*\*\*\*\*'). Below the password field is a list of criteria: '✓ 8+ characters', '✓ lowercase letter', '✗ uppercase letter', '✗ number', and '✗ special character'. A black 'Submit' button is at the bottom right of the form box.   
2. The second mockup (middle) has a similar layout but with the 'Password' field above the 'Email' field. The password field is empty, and the criteria list is identical to the first mockup.   
3. The third mockup (right) shows a more compact design. The 'Email' field contains 'test@test.de' and the 'Password' field contains '\*\*\*\*\*'. The criteria list is identical to the others. A black 'Submit' button is positioned below the criteria list.

Please find an inspectable version of the design [here](#).

## Implementation

- Use whatever technology that helps you to achieve a good result, but **try to avoid unnecessary dependencies**.
- Structure styles with an eye on **scalability & maintainability**. Please avoid using CSS libraries (e.g. Bootstrap) – we would like to get an impression of your way dealing with styles.
- Please create a repository and deliver your code on Github. Make **regular commits** to document your workflow and progress.
- **Have fun!**

## Evaluation criteria

Our goal is to get an impression of the way you approach problems and their solutions. We consider the following criteria to evaluate your solution.

- **Handover:**
  - Is the solution delivered via Github repository?
  - Are there regular commits to document the development process?
  - Does the repository contain a readme? Does it explain how to use your solution?
  - **Bonus:** Does it give rationale about the selected technologies?
- **UX / UI:**
  - Does the solution adapt well to different viewports?
  - Does the solution match the design requirements?
  - Does the solution have a decent level of accessibility?
- **Code Quality:**
  - Is the code clean, consistent, well-structured and easy to understand?
  - Do you demonstrate a good understanding of ES6 & semantic HTML?
  - Does the project structure scale well when being extended?
  - Does the repository only contain the required files?
  - **Bonus:** Are code quality tools in place?
  - **Bonus:** Are there unit tests for components and business logic?
- **Best Practice:**
  - Are the choices of libraries etc. appropriate for the given task?

## Asking questions is good

In our team we appreciate engineers spending time discussing with colleagues about the challenging details of a problem as well as potential solutions. Usually this leads to more efficient development and higher quality of the results.

That is why we are happy to help and will not penalize you for asking questions.