Full Stack Developer - Coding Test

OnlyCoders.net

This coding test aims to assess the skills and expertise of JavaScript developers by providing them with a set of programming challenges. The test is designed to evaluate the developer's proficiency in JavaScript programming concepts, algorithmic problem-solving ability, understanding of coding best practices, and their familiarity with relevant libraries, frameworks, and tools.

**Duration:** The test includes a series of coding challenges that need to be completed within 2 weeks since you received this test.

**Instructions:** The developer will receive a comprehensive list of coding tasks divided into three levels according to their complexity. Each level entails specific requirements, input/output specifications, and constraints that must be followed. **The developer is expected to create code that is not only clean and efficient but also thoroughly documented, adhering to the instructions provided for each stage.** It is mandatory for the developer to complete at least level 1 in order to submit this coding test. Completion of level 2 and 3 will be considered as a bonus achievement.

**Submission:** To maintain a systematic record, the finished test must be submitted as a Git repository. Following the submission, our team will take approximately 2 weeks to carefully review your work and determine if it aligns with our expectations. We will contact you promptly thereafter.

**Evaluation Criteria**: The code submitted by the developer will be assessed based on the following factors:

1. **Correctness:** The solutions should produce the expected outputs for a variety of test cases.

2. **Efficiency:** The code should be optimised and performant, taking into consideration time and space complexity.

3. **Readability:** Clear and well-structured code with appropriate variable/function naming conventions, comments, and documentation.

4. **Modularity and reusability:** Designing code that is modular and easily maintainable or extendable.

5. **Compliance with coding standards and best practices.**

Overall, this coding test aims to assess the JavaScript developer's competence, problem-solving skills, and ability to apply JavaScript concepts to real-world scenarios. It provides an opportunity to showcase their expertise and demonstrates their potential suitability for JavaScript development roles.

# Tasks

## Level 1

To showcase your expertise in HTML5, CSS3, and JavaScript (ES6) and create a sample page based on the attached sample \*1, please ensure the following:

1. Pay attention to detail: Make sure that every element on the page is accurately represented and mimics the design in the attached example. Pay close attention to spacing, typography, and alignment to provide a polished and professional appearance.
2. Mobile-friendly design: Ensure that the sample page you create is responsive and adapts well to different screen sizes and orientations. Test the page on various devices and browsers to ensure it looks and functions flawlessly.
3. Create nice animations in javascript. Use animation libraries, optimise performance, leverage hardware acceleration, pay attention to timing and easing functions, create responsive animations, add interactivity, combine animation techniques, test on various devices and browsers, etc..

**Bonus Points:**

1. **Bootstrap integration:** As an added bonus, incorporate Bootstrap into your implementation to optimise the page's responsiveness and utilise its built-in components and grid system. Leverage Bootstrap's CSS classes to maintain consistency and improve the overall user experience.
2. **ReactJS integration:** If possible, utilise ReactJS to develop the sample page. This demonstrates your proficiency in using modern JavaScript frameworks. Consider utilising React's virtual DOM and component-based architecture to enhance the page's performance and maintainability.

Remember to adhere to best practices for HTML5, CSS3, and JavaScript, such as using semantic markup, modular CSS, and ES6 features. Comment your code appropriately to provide clear explanations of your implementation approach in the README file.

## Bonus Level 2

To showcase your skills with Next.js, I'd like you to create a simple form that includes fields for email, password and confirm password. The purpose of this form is to collect user information, which will be sent to the backend for validation. If the information is valid, the backend should return a success message; otherwise, an error message should be displayed to the user.

1. In your project, create a form component that includes input fields for the email, password and confirm password.
2. Set up a form submission handler in the component. When the form is submitted, the handler should collect the values of the email, password and confirm password fields.
3. Use the built-in form validation features in the front-end to ensure that all fields are filled out correctly. Display any necessary error messages if the validation fails.
4. Once the form is validated, make an API call to the backend to send the collected user information.
5. In the backend, validate the received email, password and confirm password. If the information is correct, return a success message; otherwise, return an error message.
6. Display the returned success or error message to the user on the frontend.

## Bonus Level 3

To demonstrate your knowledge of MongoDB, store the email and password provided from the form in MongoDB.

# 

# Attachments

\*1 [Sample Page](https://drive.google.com/file/d/1xQhvv3Qjqpk8fiycD5u0YdjPzCQnWJdJ/view?usp=sharing)

