+38 066 308 57 84
hr@abz.agency

Frontend Developer
Test Assignment

(!) General Information

1. Requirements and deliverables for your test assignment solution:
    a. Deploy your work on any hosting (so that it can be viewed in a browser without us deploying your source code ourselves) and send us a link to it.
    b. In your email specify the list of completed items and which tools, libraries, etc. you used in yoru solution. If you did not complete all the items of the test assignment, please clarify the reason why you have not completed them (not enough time, not enough experience/knowledge, something else).
    c. Be sure to attach to the letter the most detailed and up-to-date CV, indicating the city, photo, contacts and previous work experience (even if not related to IT).
    d. Provide a link to the github/bitbucket repository.
    e. Write in the email how many hours were spent on the assignment and which difficulties/problems you have faced.
    f. Send your email to the email address **hr@abz.agency**.

2. Materials for this test assignment:
    a. All materials used in this test assignment **were specially developed** by abz.agency employees to test applicants' knowledge as part of the test assignment for this vacancy and have nothing to do with our commercial projects.
    b. Your work will not be used for commercial and (even) non-commercial purposes.
    c. You **must not use** any materials of this test assignment and your solution to it for any commercial or non-commercial purposes (including in a portfolio, for organizing courses, or as a test in another company).

Important! *Your solution to the test assignment will only be considered by our team if it complies with **all** the requirements listed above.*

+38 066 308 57 84
hr@abz.agency

(<) Test assignment

1. Working with mockups (HTML/CSS). You will find Figma project and source code here.
    a. There is only one page with a deliberately minimized set of styles and components to keep the test assignment as short as possible.

2. Working with REST API (GET). You will find API documentation (OpenAPI) here.
    a. Implement the “Working with a GET request” block according to the mockup and API documentation. Display 6 users on the API request result page. The "Show more" button should be hidden when the last page of API query results is reached. Users are sorted by registration date (the newest first).
    b. To display radio buttons on the registration form, use the GET /positions method from the API documentation.

3. Working with REST API (POST) – registration form block “Working with a POST Request”
    a. Implement front-end validation in accordance with mockups and API documentation.
    b. Implement the business logic of the registration form in accordance with mockups and API documentation.
    c. After successful registration, update the list of users in the “Working with a GET request” block. If the “Show more” button has been clicked (i.e. more than one page of users has been loaded from the API), collapse all and display only the first page of the result of the GET request. As a result, the new user will be displayed first and you will be able to check the correctness of the form without reloading the page.

4. Website optimization (bonus task). Minimize and optimize css, js, images, etc. To do this, you need to deploy your work on any hosting available to you and send any domain available to you to it.
    a. Check your work using Google Page Speed and make sure your work is in the green zone.
    b. Check your work using Google Chrome Performance Audit / Lighthouse and make sure your work is in the green zone for Performance, Best practices, SEO (mobile and desktop for 3G).
    c. Check your work using Webpagetest and make sure the scores are close to AAAAAA.

+38 066 308 57 84
hr@abz.agency

(T) Technical requirements

- React / Vue;
- Responsive / HTML5 / CSS3;
- CSS preprocessor (Sass/Less) or PostCSS;
- Organize your CSS styles in a readable way (group CSS by functionality, write explanatory comments, split styles into multiple files by functionality, etc.);
- Pay attention to the mockups and style guide when working with the user block. The assignment requires your attention to the details of the content output. Details like number of blocks, "...", tooltips, line breaks, etc.
- Markup must be pixel-perfect
- Latest browser;
- Self testing in the following browsers (you can use BrowserStack):
    - Chrome, Firefox, Edge, Safari (Windows);
    - Chrome, Firefox, Safari (MacOS);
    - Chrome, Safari (iOS);
    - Chrome (Android).

Important! *We will pay attention to how clean your approach to writing CSS and Javascript code is. You can use any third party CSS and Javascript libraries without any restrictions. Third party CSS/Javascript libraries must be added to the project via npm/yarn dependency managers (not copy-pasted). Compiling the project must be done with npm/webpack.*