ahead& take-home Kotlin & Vue assessment

Full stack app

Task 1

Your task is to create a small full stack dummy application with a backend written in **Kotlin with Spring Boot framework** and a frontend written in **TypeScript, Vue.js with Vite framework.** The frontend should communicate with the backend via a REST API.
There are many ways to implement this. Please try to implement it as clean as possible. Keep the following terms in mind: separation of concerns, DRY, clean code.

Backend

1. Create a private repository on GitHub
2. Initialize a Spring Boot project in it with **Java 17, Kotlin programming language and Gradle**
3. Describe the commands necessary to run the project in the README
   Note: Running the project should be “editor-agnostic”, so it shouldn’t require any particular editor like Eclipse or Intellij IDEA to run
4. Write a class representing a user. It should have the following fields: firstName, lastName, email, id
5. Write an interface representing user storage. It should describe two methods: findAll(): List<User> and save(user): User
6. Write an implementation for this interface. The implementation should store the users in the RAM
7. Create a users controller with endpoints “GET /users” - to get all users and “GET /users/random” - to generate a random user, save to the user storage and return the created user

Additional notes

- No persistence is required between the runs of the project, since the users are stored in the RAM
- Do not return the user model directly. Instead return a DTO with the same fields
- The users controller shouldn’t know what implementation of the storage is injected

Frontend

1. Create a private repository on GitHub
2. Initialize a Vue.js project with Vite
3. Describe the commands necessary to run the project in the README
4. Write a simple UI that communicates with the backend which you created in the previous task and calls its two endpoints
    a. The UI should list all users
    b. The UI should have a button to generate a random user
        i. When the button is clicked, the user list should be reloaded without reloading the page

Task 2

Backend

1. Create an auth controller that has the following three endpoints:
    a. /register
        i. Accepts first name, last name, email
        ii. Registers the user utilizing the user repository
    b. /request
        i. Accepts an email
        ii. Creates a 6-digit verification code in the RAM
        iii. Outputs the code in the console on the backend
    c. /login
        i. Accepts an email and a 6-digit code
        ii. Checks the code against the one in the system (created in the /request endpoint)
        iii. Returns 200 status code, user ID, first name and last name if the authentication was successful (if the code matches)
        iv. Returns 401 status code if not successful (if the code doesn't match)

Frontend

1. Implement a functioning /register page
2. Implement a functioning /login page
3. After login, save the user data in local storage and display the user initials in the header

Additional notes

Task 2 doesn't have strict requirements, you define the architecture yourself. The process just has to function.

How to submit

Submit the project by inviting https://github.com/eugen-bondarev to both repositories (frontend and backend).

How we evaluate

We're going to run both projects simultaneously and check the functionality end-to-end as well as the code behind it.