Full Stack Test Task

Task description

Implement a module that allows a user to sign up and sign in to the application. The application should be production-ready and adhere to industry best practices for both front-end and back-end development.

Front-end development:

Requirements:

- Develop the user authentication module using either the React or Vue framework.
- Design elements are open to your creativity.
- You must use TypeScript.
- You have the flexibility to choose any additional modules or libraries (including design frameworks) if necessary.

Sign up page:

Create a signup form with the following fields:

- Email: Valid email format validation.
- Name: Minimum of 3 characters.
- Password: Password requirements are as follows:
  - Minimum length of 8 characters.
  - At least one letter.
  - At least one number.
  - At least one special character.

Sign in page:

Create a sign-in form with fields for:

- Email
- Password

Application page:

Create a page that displays the following:

- A welcome message: **"Welcome to the application."**
- (Optional) Add a logout button to end the session.

Back-end development:

Technical stack requirements:

- Implement the back-end endpoints using the NestJS framework and integrate MongoDB as the database.
- Add at least one protected endpoint
- Add a readme file with basic information on how to work with the repo.
- You can choose the appropriate ORM and other libraries if needed.

Notes:

Build API endpoints to sign up and sign in users to the application, taking in the account requirements to the fields described in the Front-end part.

Nice to haves:

- Implementing logging on the back end
- Following best practices for security
- Api documentation

Submission:

Once completed, create a public GitHub repository and push your code to it.
Share the repository link with the recruiter.

Scoring Criteria

Submissions will be evaluated based on:

1. **Functionality:** Does the application meet the requirements?
2. **Production-Readiness:** Is the code secure, and maintainable?
3. **Code Quality:** Is the code clean, modular, and easy to understand?
4. **Bonus Points:** For implementing optional features like Logging, Error handling, testing, basic CI/CD or API documentation.