Vue3 test assignment

This is a test assignment designed for a middle-level Vue3 developer. The estimated time required to complete this assignment is between 8 to 12 hours.

Test Assignment:

1. Create a new Vue3 application using `vite` as the bundler. Integrate `vite` with `vitest`, a coverage reporter, and choose between `vue-test-utils` or `vue-testing-library` for testing. Utilize TypeScript. Following TDD as must.
2. Implement authentication and registration using an OpenID provider Auth0, with the `auth0-lock` library.
3. Define two roles for users: admin and user.
4. During registration, use a parameter from the query string to assign a role to the registered user. This parameter should be stored in Auth0 for future role determination.
5. The application should feature two layouts: one for authorized users and one for unauthorized users.
    - The layout for unauthorized users should include an Auth0 authentication form, and all other routes of the application should be protected, inaccessible to unauthorized users by any means.
    - The layout for authorized users should include a page displaying a list of notes:
        - Each note should display:
            - Creation date
            - Author username / email address
            - Note text
            - Delete note button (available only to the author and users with the administrator role)
            - Edit note button (available only to the author)
            - If edited, the note should display the date of its last modification.
        - Utilize LocalStorage as a temporary database for storing notes.
        - Place a note creation form at the bottom of the page. The form should have only one field for the note text and should include validation:
            - It cannot be empty.
            - A user cannot have two notes with identical text.
            - If the field value is invalid, it should be highlighted in red, and a message indicating the violated validation rule should appear below the field.
        - The page should be scrollable to accommodate all notes. Notes should be arranged from bottom to top, from the earliest creation date to the latest, if the note has no edit date. If there is an edit date, such notes should rise to the top and be sorted by the edit date.
        - All notes should have the same width and be centered on the page. The height of the note should be adaptive, adjusting to the content.
        - Place a header at the top of the page with a logout button aligned to the right. Upon clicking the button, the user should be logged out and returned to the layout for unauthorized users.
        - The list of notes should be reactive. There should be no need to manually refresh the page to see changes upon deleting/adding a note.
    - Users with the administrator role should have access to all notes.
    - Users with the user role should only have access to their own notes.
6. Follow the BEM methodology when styling the page.
7. Implement the Single File Component (SFC) approach. Components should be as independent as possible from each other.
8. The written code should be covered with unit tests. The overall line coverage should be no less than 80%.

Evaluation:

Evaluation will be based on the following criteria:

- Code quality and architecture according to best practices.
- Implementation of core application functions.
- Visual appearance and user interface.
- Organization and structuring of Vue 3 components.
- Application of modern development practices and standards.
- Presence and quality of unit tests.

Additional points may be awarded for:

- Additional functionality that enhances the user experience.
- Implementation of animations and transitions.
- Integration with an external API for data storage.
- Any other improvements or additions that you deem appropriate.

Submission Requirements:

You must provide a link to a Git repository containing all the source code of your project, along with instructions for its setup and testing.

Good luck!