

## **React Module Test | Notes App | Problem Statement:**

Your challenge is to build out a minimal note taking app.

### **Features List:**

1. User should be able to create group for notes, a popup open up to create a new group and if the user clicks outside the popup then you need to close the popup
2. Make sure that no duplicate groups are allowed.
3. The display picture should consist of two letters from the group.
4. The user should be able to make groups, if the length of the name given to the group is less than 2
5. User should be able to add notes in a particular group
6. User should be able save the note by pressing Enter key or clicking on the Enter icon in the text input
7. You need to save the notes in any browser storage you find relevant for this task so that every notes and group should persist on page reload
8. As soon as the user changes the group, you need to fetch all the notes related to that particular group
9. You also need to save meta data like date and time when the notes were saved and last updated.
10. DO NOT change the design of your application and keep it as close as possible to the design that has been provided to you.
11. The send button only changes color when there is content in the text box, as per the design given.

NOTE: If you have any queries regarding the problem statement, feel free to ask in the chat box given on your dashboard.

## **SOME IMPORTANT INSTRUCTIONS FOR MAKING PROJECTS -**

### **Do's:**

1. Always use raw/vanilla css
2. Always use modular css
3. Always structure your project properly
4. Always use proper naming for variables and functions
5. Always use some kind of basic validations for forms
6. Use jwt auth mechanism
7. You can use any toast library for showing toast. Design can be flexible for toast.
8. Your password should always be stored in encrypted form, always prefer any encrypting library to do that. Also password comparison should be done use the same encryption library
9. Proper error handling should be there on the backend and frontend. Your backend should return proper response in case of anything goes wrong

### **Don'ts:**

1. Don't use any css library or frameworks
2. Don't store your passwords as plain string, it must be encrypted
3. Don't implement filtering, sorting or searching on the frontend
4. If your code is found to be copied from anywhere then your solution will not be considered for the evaluation

### **Evaluation Criteria:**

1. Implementing pixel perfect UI
2. Coding standards and best practices
3. All the functionality should work properly which are the part of checklist
4. How well edge cases are handled
5. How well errors and validations are done