Shippigo Feature Building (Full Stack)

Greetings Candidate,
Here is your problem statement for the first round of the test.

Tech Stack:

- MongoDB/SQL as the database
- NodeJs as the server side runtime
- Express as API framework
- ReactJs / Angular as frontend framework / library

Task:

You are expected to create a fully functional todo app. Below are the details that need to be considered.

1. Create a registration page . Users will be able to register using either google SSO or using email directly. The fields that need to be available in db are
    - Email
    - Name
    - Password (Skip in case of google registration)
    - Role - Use by default user (only 2 are possible user and admin)
    - Registration date
2. SignIn page. It will either use email and password or google sso
3. After login by default the user will see all his tasks created in the last 15 days.(only heading of the todo task.). Use different colors or some other highlighter for different status.
   Eg: - Pending tasks can be in green , completed can be in red , onHold tasks can be in yellow. Or you can use strikethrough for completed tasks. *This is just an example, you can use your own ways*
4. User can filter his/her tasks based on creation date, status (pending ,completed, on hold)
5. Details of a particular task - User will be able to click on a particular task and can see the details of the task , including the file also. (Provide a clickable button for file and download the file on click)
6. Create a Todo task. Users can create a new task. Which will have at least below mentioned fields. You can use more according to your requirements.
    - Heading
    - Description
    - File - optional field. Upload a file to AWS S3 and save the link
    - Status - By default it should be pending.
    - Add others according to your requirements.
    - Creation date
7. Edit task- User can edit details of the task which has been created earlier. Editable fields are - Heading, Description, File and status of task (mark as complete or on hold).
8. Delete Task - User can be able to delete a task which he/she has created.
9. Logout Button - Which logs users out.

10. Create an admin login also. If the admin logs in he can see all the tasks that have been created by anyone. He can also filter based on users along with other filters mentioned above.

Key Points and submission process

- User should be able to only see his tasks. No user can see tasks created by any other user except the admin
- Try to complete things mentioned above. If unable to implement any feature then skip it and complete all others.
- Create a Readme file for the project having some details about the project
- Use some sort of differentiation in displaying different types of statuses.
- Use **Morgan** to log all requests on a file. (Optional. Plus point if executed)
- Upload the project on github.
- Deploy your app. (Can use Heroku) and provide the url in readme file
- Make a 2-5 mins video giving a demo and explaining your code and share the link (Gdrive) in the readme file.
- Share the deployment url and github link via mail

Evaluation Criteria

- The tasks should match the instructions given as closely as possible.
- Code cleanliness and structure.
- Responsive UI.
- Use of typescript would be commendable.
- Ability to explain code.
- Commit frequency and relevance.

NOTE - IF ANYONE IS ONLY COMFORTABLE WITH ONE SIDE OF PROJECT (Frontend / Backend) THEN GO AHEAD WITH EITHER FRONTEND ONLY OR BACKEND ONLY. Mention it in mail.

Any issues during the assignment, please report quickly to ankur@shippigo.com (frontend) or rajat@shippigo.com (backend).

Share the completed detail over mail to kabindra@emizainc.com and nitish@shippigo.com

Best of Luck!
Team Shippigo