Frontend Evaluation

Summary

Considering you are trying to build a front-end app for the application which has the option of creating a User Story (i.e ticket or card) which contains information about what kind of task needs to be performed and then the admin will review the User Story, make changes if required. Admin will be able to approve or reject the User story created by the user. Assume there is already a backend application running and you are building the frontend application which talks to backend application by ReSTful web services (Axios).

Context

User Roles: The application will have 2 different user roles.

- User
- Admin

Each role has its own set of functionalities.

Tasks

Login as admin or normal user: Create a login page with following components:

1. Email input field
2. Password field
3. Submit button
4. A toggle button to login as admin or normal user

On the click of submit button make an api call to login. Here is the apis

Base Url: https://test-archimides.free.beeceptor.com/

Login as User:

Method: POST

EndPoint: ${base url } + /api/login

Login as Admin:

Method: POST

EndPoint: ${base url } + /api/admin-login

Response format
{

userRoles: ['User'],

firstName: 'Alpha',

lastName: 'User',

id: 2

}

Once logged in based on the userRoles it should navigate to either create a story option when logged in as user, else if logged in as admin should see the stories available for review
NOTE: When admin login is checked it should not allow users to login and show error. Also basic form validation needs to be implemented

Creating a Story as user: Once logged in, As a user of the platform I should be able to create a User Story i.e a ticket or card which will contain the information about the task which needs to be performed by the developer. A user story typically will contain :

- Summary (Input field)
- Description (Text area)
- Type (like enhancement, bugfix, development, qa etc) (Dropdown)
- Complexity (Dropdown with values: Low, Mid, High)
- Estimated time for completion (Input)
- Cost associated to it (numeric Input showing $ as prefix of input)

Create a page with form having the above inputs controls and a **SUBMIT** button. On Submit click **validate** the form & make the following api call

Base Url: **https://test-archimides.free.beeceptor.com/**
POST /api/createStory
Payload: Send the form values as payload

Once the story is created then navigate the user to story list

View stories created by user: Create a list page with the following columns to show the stories created by a user:

- Summary
- Description
- Type
- Complexity
- Estimated time for completion
- Cost associated to it

Only the user who created the user story should be able to view it apart from admin. Users will not be able to see each other's user stories. Use field `*createdBy*` to filter

API to get stories

Base Url: https://test-archimides.free.beeceptor.com/
GET api/getStories
Dummy response:

```
[{
createdBy: 2,
summary: '1st story created by 2',
description: 'dummy desc',
type: 'enhancement',
complexity: 'high',
estimatedHrs: 1,
cost: 100
}]
```

View list of created stories as Admin for approval: When logged in as **Admin** admin should be able to see all the user stories created irrespective of ***createdBy*** , Admin should be able to see the list of user stories created by users to review. The story list will be the same as the one that the user sees. It will have similar columns

- Summary
- Description
- Type
- Complexity
- Estimated time for completion
- Cost associated to it
- Status

NOTE: If for a record the status is ‘rejected’ then row color should be red, if the status is accepted then row should be in green else the row color should be black

API to get stories

Base Url: https://test-archimides.free.beeceptor.com/
GET `api/getStories`
Dummy response:

```
[{
createdBy: 2,
summary: '1st story created by 2',
description: 'dummy desc',
type: 'enhancement',
complexity: 'high',
estimatedHrs: 1,
cost: 100
}]
```

Reviewing as admin: As an Admin, When clicked on a row on the user story list it should take to a screen showing the details of the record and with options to Approve or Reject the story. Admin can either approve or reject the user story.

Elements on Page

1. Read only fields for the following fields
   a. Summary
   b. Description
   c. Type
   d. Complexity
   e. Estimated time for completion
   f. Cost associated to it
   g. Status
2. Two buttons Accept(in green color) and Reject (Red color)

After the story is approved or rejected, then update the status of the record on the table and navigate to the story list.

Process

a. Please design the UI as per what you feel is right.
b. Design should be responsive and work on both desktop and mobile browser.
c. All pages should have separate routes and when we refresh the browser, it should remain on the same page if authenticated.
d. On App launch you need to check if the user is logged in. If not logged in, then show the login page.
e. Manage the base url in a separate config file.
f. Use Axios for making api calls.
g. Usage of redux is encouraged but not mandatory.
h. Please upload the working copy of your code to GitHub and send us the link .
i. Please state all your assumptions in the readme file of your GitHub repository.

As a Bonus, it will be highly appreciated if you can also provide Javascript based unit tests for your front-end code.

POSTMAN COLLECTION:

You can import the postman collection and check the apis incase you find any issue integrating the apis - Archimydes-Evaluation-apis.postman_collection.json