

### Question 3

**Problem Statement:** Andrew Parker is the Project Manager at ABC Corporation. From last year the number of employees is increasing rapidly in his company and now it is becoming difficult to manage the resources. So he needs a very basic resource management system by which he can easily track that who is available and who is not.

He has provided the required HTML and Css files (attached). You have to use them in your React/Vue/Angular application.

Let's go through his requirements.

Requirement analysis:

There is basically one page which is the Dashboard. Below is the prototype image of it. This page has two sections. The left side is for getting an overall view for the total employee and available employee number and employee is will be added by clicking the Add Employee button, the right side contains the list of the employees, I believe that it is not a good practice to load all the employee for the very beginning, might be there will be a pagination or load on scroll.

It will be cherry on the top if you'll provide a search by text( it might be by name or by designation), oh! I forgot to tell you one thing, could you show me the available employees at first?

And from the list, I can change the availability. After changing the availability, the list won't be shuffled, but if I refresh the page then the employees' position must be shuffled.

In the dashboard there is an add button by which an employee can be added.

On clicking on Add Employee, the following modal will be opened.

I just need a very basic validation form to add an employee. Like -

- i. All the fields are mandatory
- ii. Age must be an integer

After pressing the save button, if the employee is added successfully then a popup will come that employee is added successfully, if not then the same popup will come with a specific reason that why the employee is not added. If the employee is successfully added then the modal will be closed and the newly added employee will be shown at the top of the employee list and the total number of employees, as well as the available number of employees, will be increased.

On each row of the employees' table, there will be a edit button (pencil icon). On clicking on edit button of a particular employee a modal should get opened. The modal should look like the same as the previous one (You can reuse the same). Only all the available values will be pre-filled. From this modal an employee details can be updated.

Also on each row of the employees' table, there will be a delete button (trash icon). It will be used to delete that particular employee.

I forgot to tell you one thing, my company has mainly four departments - Frontend Development, Backend Development, Testing, and Deployment.

Note:

Store all the data locally (local storage/session storage)

If you find some special cases / action flow not mentioned above, feel free to implement your own approach in such cases (For example in the employee delete feature, Andrew has not mentioned about any confirmation prompt before deleting. You can implement your approach in such cases).