QUESTION 4

Building a Basic Resource Management System

Problem Statement:

Andrew Parker, the Project Manager at ABC Corporation, needs a basic resource management system to efficiently track employee availability. The provided HTML and CSS files are intended for use in your React/Vue/Angular application. Let's delve into the requirements.

Requirements Analysis:

1. **Dashboard Overview:**
   • The Dashboard consists of two sections: left side for an overall view of total and available employees, and right side listing employees.
   • Employees can be added by clicking the "Add Employee" button.

2. **Employee Listing:**
   • The employee list may incorporate pagination or load on scroll to optimize resource utilization.
   • A search functionality by text (name or designation) is preferred.
   • Initially, show available employees first.

3. **Availability and Shuffling:**
   • Availability of employees can be changed from the list.
   • Changes in availability won't shuffle the list; however, a shuffle occurs upon page refresh.

4. **Add Employee:**
   • The "Add" button opens a modal with a basic validation form.
   • Mandatory fields must be filled, and age must be an integer.
   • After saving, display a popup indicating success or a specific reason for failure.
   • If successful, close the modal, and the new employee is shown at the top, increasing the total and available employee count.

5. **Edit Employee:**
   • Each row has an edit button (pencil icon) opening a modal with pre-filled values for updating employee details.

6. **Delete Employee:**
   • Each row has a delete button (trash icon) for removing an employee.
   • Implement your approach for cases like confirmation prompts, which Andrew hasn't explicitly mentioned.

7. **Department Information:**
   • The company has four departments: Frontend Development, Backend Development, Testing, and Deployment.

Implementation Details:

• Store all data locally using local storage/session storage.

Note:

If you encounter special cases or action flows not covered above, feel free to implement your approach. For example, in the employee delete feature, where Andrew has not specified any confirmation prompt before deleting, you can implement your approach in such cases.

Instructions:

1. Utilize the provided HTML and CSS files for your application.
2. Implement the specified features using React/Vue/Angular.
3. Incorporate local storage/session storage for data persistence.
4. Feel free to address any additional scenarios or flows not explicitly mentioned.

Submission:

Submit the code for your React/Vue/Angular application, ensuring it meets the outlined requirements.