

ALGORITHM

1. Start the application.
2. Define the necessary routes in the **app-routing.module.ts** file to navigate between different components.
3. Create the database JSON file (**db.json**) with an initial list of employees.
4. Implement the login functionality in the **LoginComponent**
 - Bind the input fields to the **username** and **password** properties using two-way data binding.
 - On form submission, check if the entered username and password match the predefined admin credentials.
 - If the credentials are correct, navigate to the main employee management page (**/employees**).
 - If the credentials are incorrect, display an error message.
5. Implement the employee management functionality in the **EmployeeManagementComponent**:
6. Fetch the list of employees from the JSON database using the **HttpClient** in the **ngOnInit** method.
7. Display the list of employees in a table using ***ngFor** directive.
8. Provide options to add, view, update, and delete employees.
9. Implement the add employee functionality in the **AddEmployeeComponent**:
 - Create a form to enter the employee details (first name, last name, email).
 - On form submission, send an HTTP POST request to add the new employee to the database.
 - Clear the form and refresh the employee list.
10. Implement the employee details functionality in the **EmployeeDetailsComponent**:
 - Retrieve the employee ID from the route parameters.
 - Fetch the employee details from the JSON database based on the ID.
 - Display the employee details in a table.
 - Provide options to go back, update, and delete the employee.

11. Implement the update employee functionality in the **UpdateEmployeeComponent**:

- Retrieve the employee ID from the route parameters.
- Fetch the employee details from the JSON database based on the ID.
- Populate the form fields with the current employee details.
- On form submission, send an HTTP PUT request to update the employee details in the database.
- Navigate back to the employee details page.

12. Implement the delete employee functionality in the **EmployeeDetailsComponent**:

- Provide a button to delete the employee.
- On button click, send an HTTP DELETE request to remove the employee from the database.
- Show a success message and navigate back to the employee management page.
- Style the components using CSS and Bootstrap classes to achieve the desired UI.
- Test the application for various scenarios and ensure proper functionality.