**1-Test Plan & Test Strategy:**

**Test Case 1: Login Verification**

Description:

Verify that the user can successfully log in.

Test Steps:

- Navigate to <http://localhost:8080/employees.html>.

- Check that the login is successful.

**Test Case 2: Tree Grid Check**

Description:

Check if the header elements are present and contain the expected values.

Test Steps:

-Navigate to the application.

-Check the existence and values of the header elements.

-Check the elements for rows (checkboxes , data).

-Check the pagination controls.

**Test Case 3: Employee Origin Functionality Check**

Description:

Check the functionality of selecting data.

Test Steps:

-Navigate to the application.

-Select checkboxes for the rows you want to see the results.

-Click the "View selected data" button.

-Verify that the displayed result matches the expected outcome.

-Deselect the checkboxes for the rows you want to remove from display.

**2-Identified Issues:**

**Authentication Functionality:**

**Problem:**

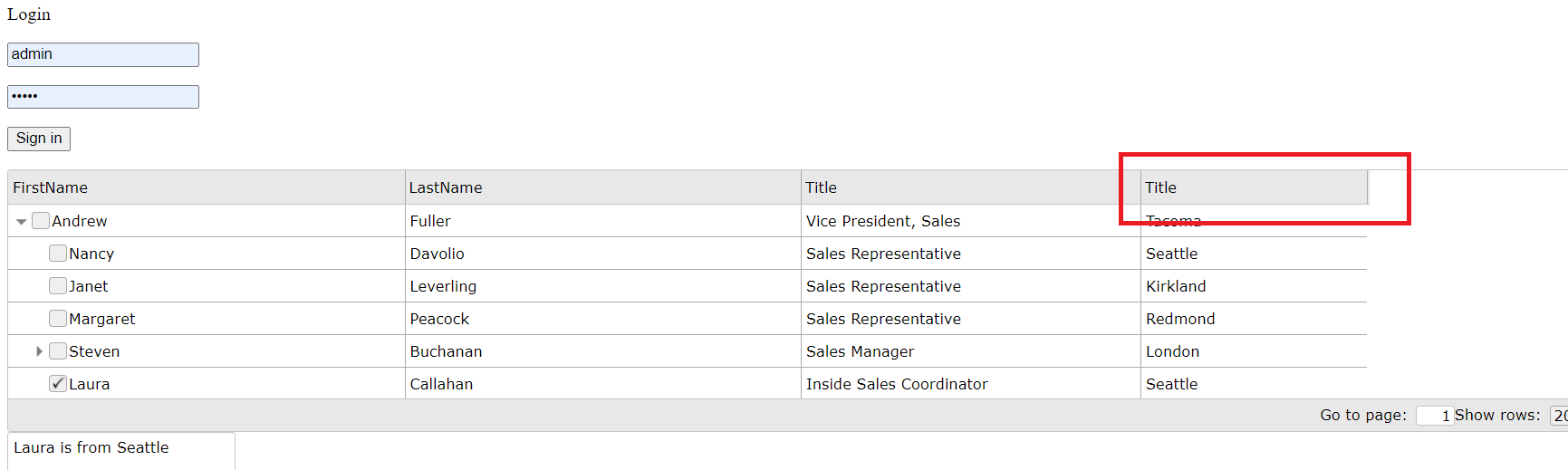
-The login is not functional**.**

-The data can be displayed regardless of the login status.

**Header Column Order:**

**Problem:**

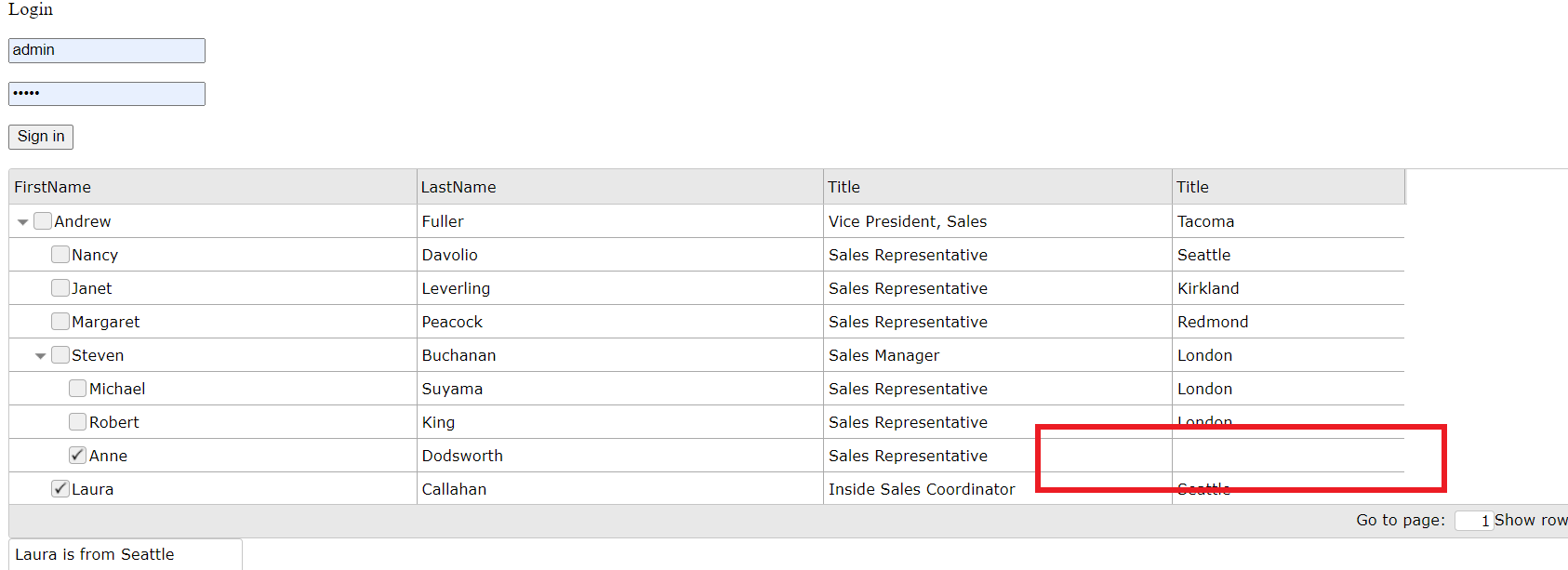
-The last column header should be "City" as per the requirement.



**Missing data for Anne’s:**

**Problem:**

-The "City" column for the "Anne" row is null.



**Pagination control not functional**

**Problem:**

-Clicking on the "Show rows" dropdown and selecting a different value (e.g., 5) doesn't update the table result as expected.

**Steps to Reproduce:**

-Choose the value 5 in the "Show rows" dropdown button.

**Expected Result:**

-The table should display the first 5 rows corresponding to the selected value.

Current Behavior : Nothing changes



**3- Acceptance criteria**

I do not agree with the acceptance criteria as it is illogical to display data without having to login, also the login should be on a different page.

**4-Test automation implementation**

Please find the implemented code on GitHub: GitHub Repository Link.

Steps to run the tests:

* Run **npm i** to install dependencies.
* Start the HTTP server using **npx** **http-server .**
* Execute **npx cypress run.**
* Execute **allure generate ./allure-results -o ./allure-report**
* Execute **allure open ./allure-report**
* View the Allure report for detailed results.

A screenshot of a computer

Description automatically generated