**Test Strategy**

**1. Introduction**

This section defines the test strategy for the Roche Employee Management application. It outlines the approach for functional, non-functional, automated, and manual testing to ensure that the key requirements—city of origin display and login functionality—are verified thoroughly.

**2. Scope of Testing**

The scope includes:

* **City of Origin Display**: Ensuring that the city of origin is displayed accurately in the employee list.
* **Login Functionality**: Verifying user authentication, session handling, and security mechanisms.
* **API Testing**: Testing the API endpoints related to employee data retrieval and login functionality.
* **Performance Testing**: Evaluating system performance under various load conditions.

**3. Test Types**

**Functional Testing:**

* **City of Origin**: Testing if the city is displayed correctly for employees.
* **Login**: Verifying that login works under all scenarios (valid, invalid credentials, session handling).

**Non-Functional Testing:**

* **Security**: Testing login for vulnerabilities like SQL injection and cross-site scripting and password encryption
* **Performance**: Measuring response times and load handling, especially during peak usage. Large employee lists might slow down the UI. Using performance testing tools like JMeter or NeoLoad to load-test the list functionality

**API Testing:**

* Validating backend API responses for employee data retrieval and login.
  + check for return status and authentication token (200 status for success)
  + check for invalid credentials (401 return status code)
  + check for SQL injection from API
  + Employee List API: Test the endpoint responsible for fetching employee data and ensure it includes the city field for successful cases
  + Check for missing/invalid fields API return response

**Automation Testing:**

* Automating test cases for city of origin display, login, and API testing to ensure coverage during regression cycles.

**UI/UX Testing:**

* Verifying that the user interface is consistent, intuitive, and error-free.

**4. Test Approach**

**Manual Testing:**

* Focus on UI/UX, validating employee data accuracy, and login flows.

**Automated Testing:**

* **Cypress**: Automating the login and employee list tests.
* **Postman**: Automating API tests for data retrieval and login validation.

**Performance Testing:**

* Using **JMeter** to test the load handling of employee lists and login functionality under stress.

**5. Risks and Mitigation**

* **Data Synchronization Issues**: Ensure accurate data between front-end and back-end by verifying APIs.
* **Security Flaws**: Ensure login encryption and protect against common web attacks (SQL injection, brute force).
* **Performance Delays**: Use load testing to prevent slow loading times for large datasets.

**Test Plan**

**1. Objective**

* Ensure that the city of origin for employees is displayed correctly.
* Verify the functionality and security of the login process.

**2. Test Items**

* **Employee List Component**: Verifying the city of origin display.
* **Login Page**: Ensuring secure and correct login behavior.
* **APIs**: Verifying backend APIs for employee data and login.

**3. Test Cases**

**City of Origin Display:**

1. Verify employee list with city of origin.
2. Validate the accuracy of the displayed city data.
3. Test empty list scenarios.
4. Verify sorting/filtering by city (if applicable).
5. Ensure UI formatting for city display.

**Login Functionality:**

1. Test valid credentials for successful login.
2. Test invalid credentials and error handling.
3. Check login form for empty field validation.
4. Ensure password input masking.
5. Test session timeout and security management.
6. Verify password encryption and data security.
7. Test password recovery (if available).
8. Ensure protection against SQL injection.

**API Testing:**

* Validate login and employee list APIs for correct data and response times.

**4. Test Deliverables**

* **Test Cases**: Manual and automated.
* **Test Reports**: Generated by Cypress and Postman.
* **Bug Reports**: Issues logged in Jira or GitLab.
* **Final Test Summary**: Report on test execution and results.

**5. Test Environment**

* Web Browsers (Chrome, Firefox, Edge)
* Backend: Node.js and PostgreSQL
* Tools: Cypress, Postman, JMeter/NeoLoad

**6. Schedule**

| **Activity** | **Duration** |
| --- | --- |
| Test Case Design | 2 days |
| Test Automation Setup | 3 days |
| Test Execution | Ongoing |
| Bug Reporting & Resolution | Ongoing |
| Final Test Report | Post-execution |

**7. Entry and Exit Criteria**

**Entry Criteria:**

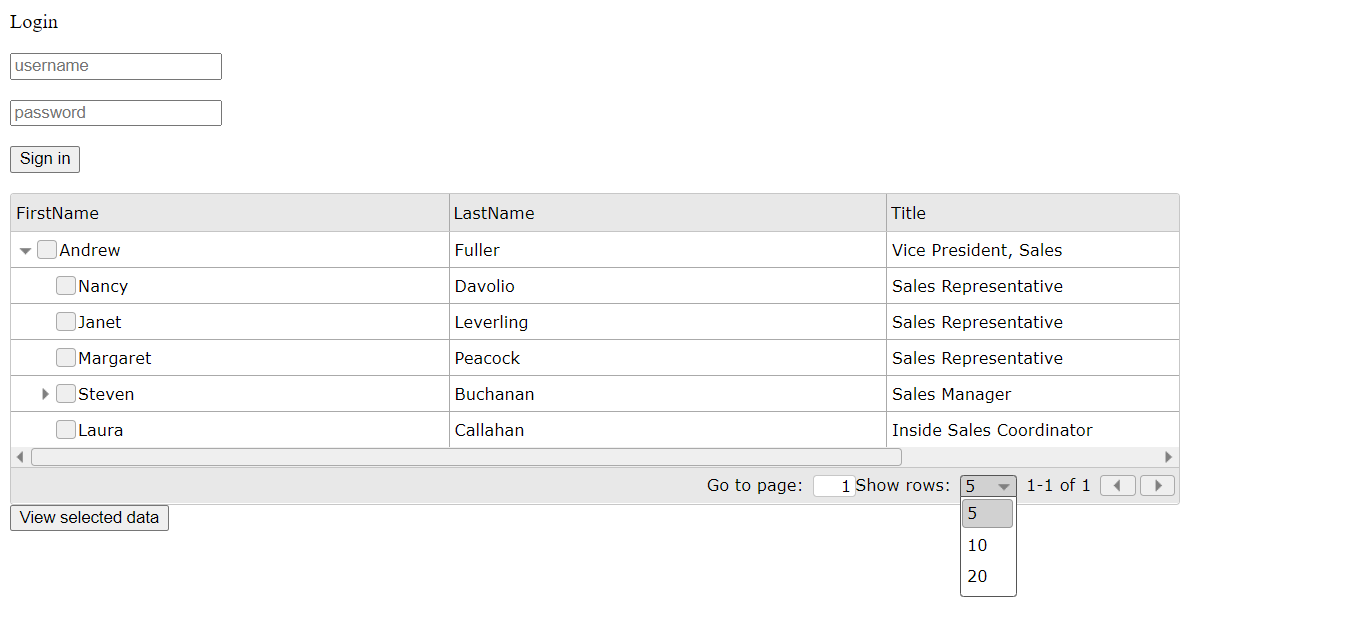
* Test cases are reviewed and approved.
* The application build is stable and ready for testing.
* The test environment is set up.

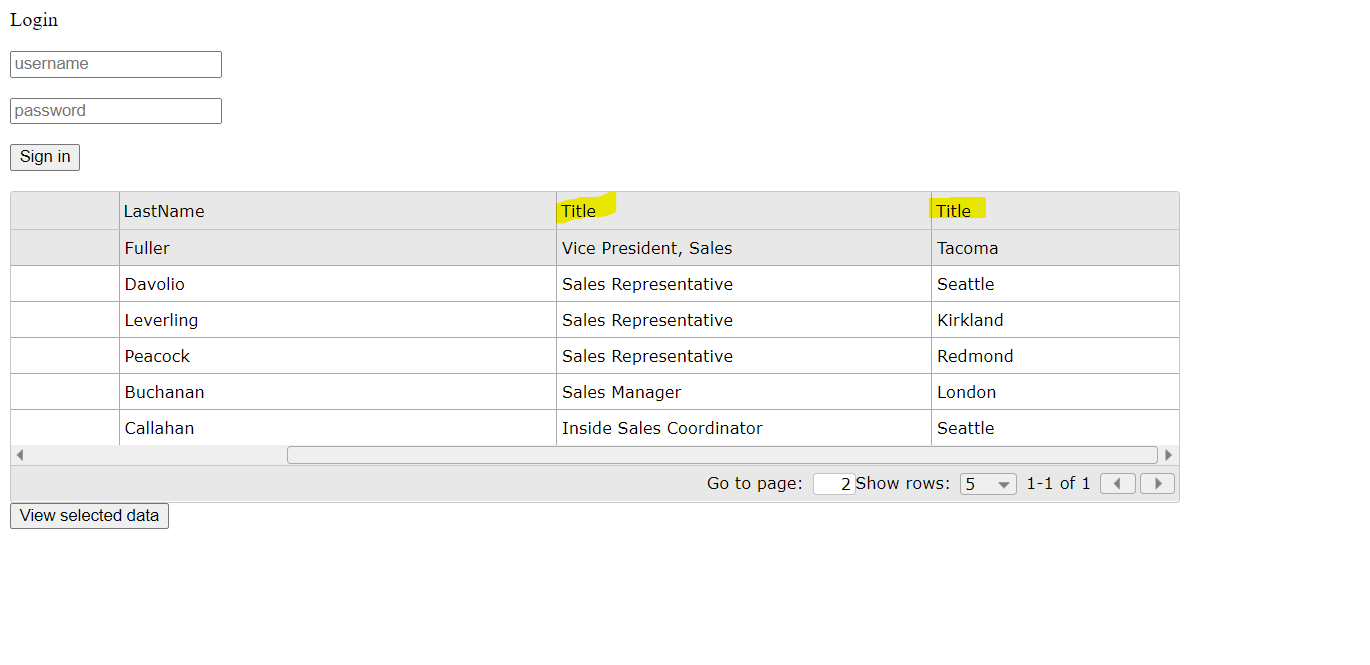
**Exit Criteria:**

* All critical test cases are executed.
* No high-priority defects are open.
* Test summary report is completed.

Identify possible issues and report one if you identify something.

1. First issue: log in functionality does not work and there is no input validation on login fields ,html element can be further enhanced to contains attribute for selectors to be more accurate. Example : cy.get('[data-cy="submit"]').click()
2. Pagination does not work : if we select a pagination of 5 the expected is only 5 element are displayed and other elements goes to next page but its not the case for this application as mentioned in screenshot more then 5 element are displayed and there “for go to page” also does not work

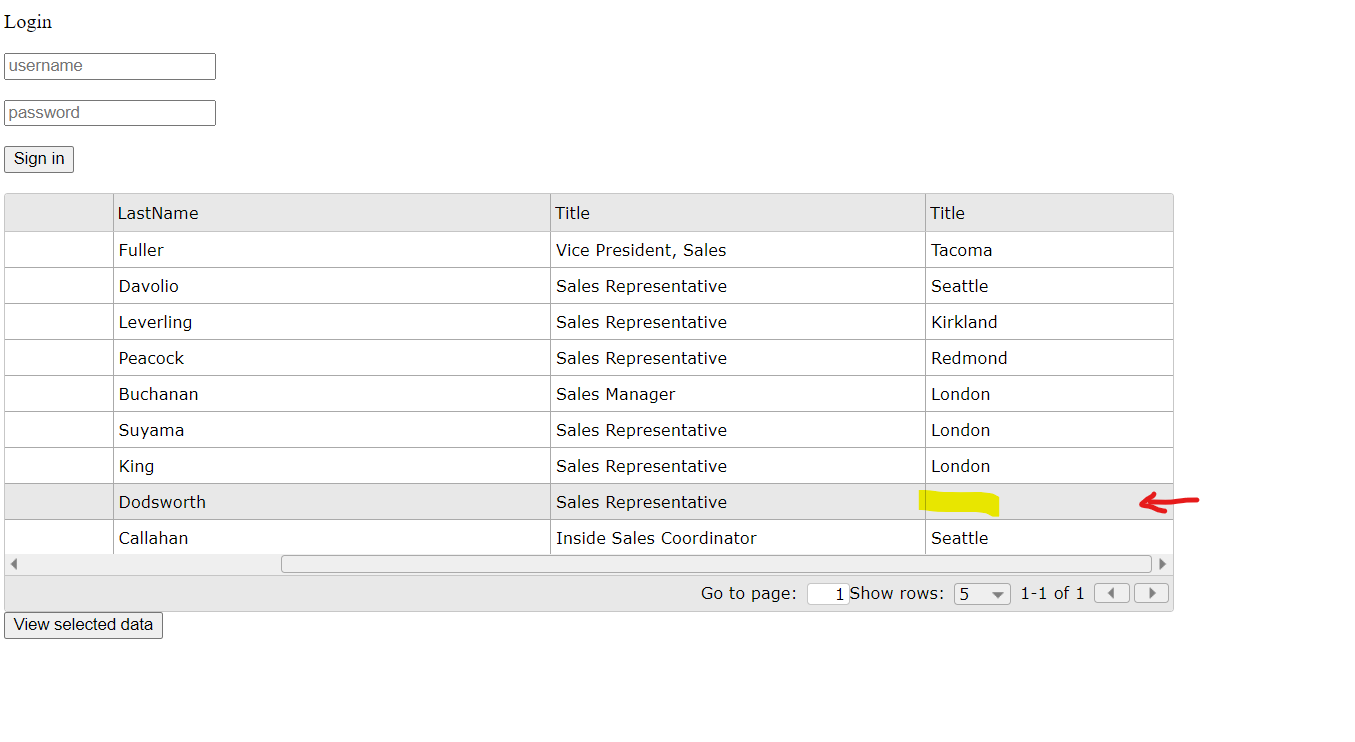


1. Title is duplicated in table ”thead” the second title should be replaced with

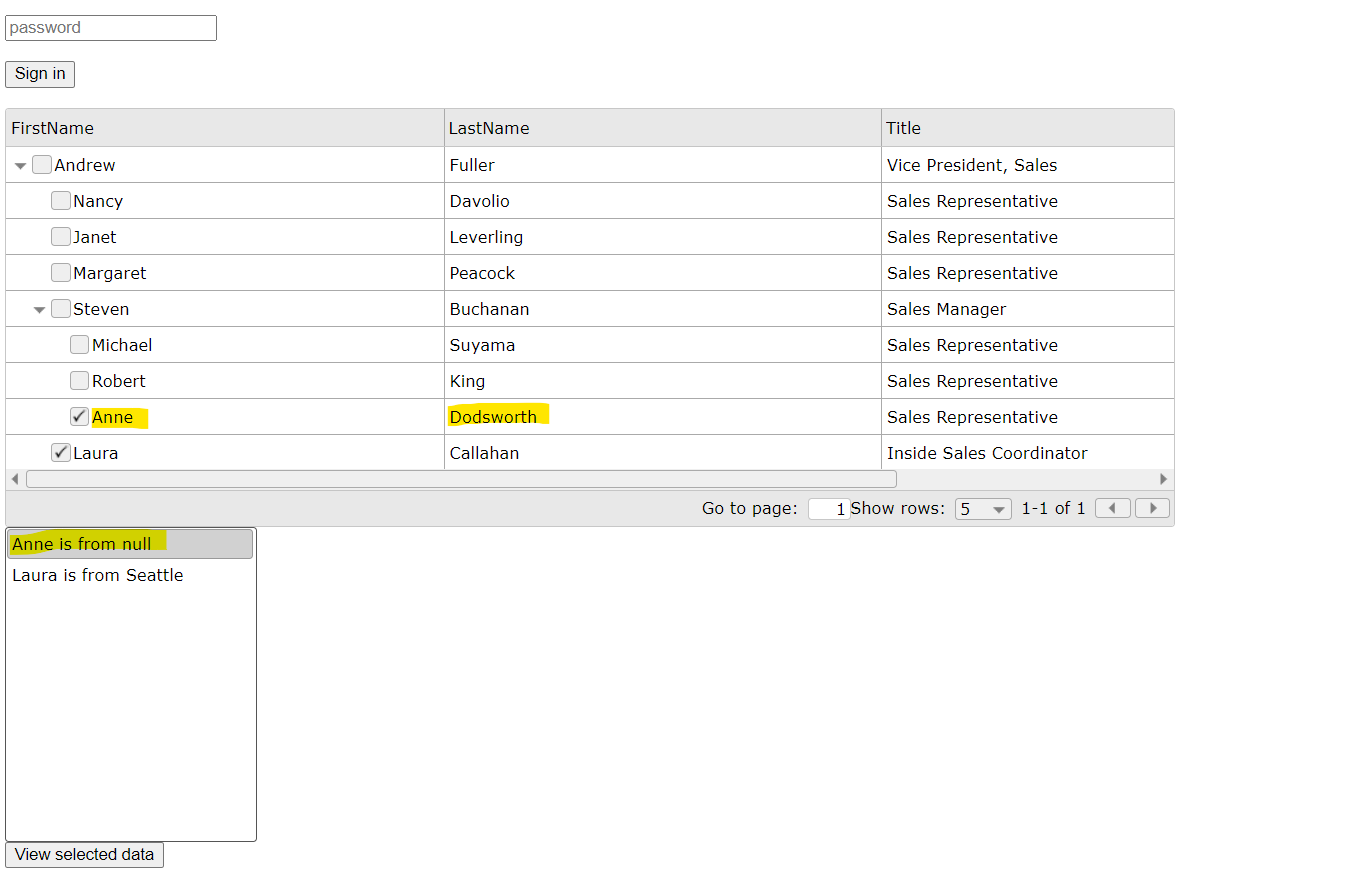
City of Origin Display

There is an empty title (should be changed to City of Origin Display)

1. field for employee name Anne Dodsworth



1. View selected data for employee Anne is displaying null as mentioned in screenshot



Do you agree with the requirement acceptance criteria? What are you missing? Put one example.

I don’t agree with the acceptance criteria because thy are missing a lot of details to be a valide such us:

**City of Origin Display functionality**

* The system must handle large datasets and display city information correctly for all employees.

**Login Functionality**

* After 3-5 consecutive unsuccessful login attempts, the system should lock the account or implement a captcha mechanism (if required for security).