Integration Test

Tested and Created by @Federico Ferracin

Integration Tests

- Per page
  - for each **.jsx** in Frontend
  - create a **.int.test.jsx**
  - cover page behavior end to end inside React app boundary

Decision

- **Unit Tests** will be done in **client/src/components/__tests__**
  - tests will live in that folder
    - small
    - isolated
    - per component
- **Integration Tests** will be done in **client/src/templates/__tests__**
  - on the different pages
    - Login
    - MainTemplate
    - UseScale

Setup

Create setup file in root for TextEncoder/TextDecoder

jest.setup.js   at root

```
1  import { TextEncoder, TextDecoder } from "util";
2  global.TextEncoder = TextEncoder;
3  global.TextDecoder = TextDecoder;
```

Update configuration file

jest.config.js   at root

```
1  /** @type {import('jest').Config} */
2  module.exports = {
3    testEnvironment: "jest-environment-jsdom",
```

```
 4
 5    // for integration test text encoder / decoder
 6    setupFiles: ["<rootDir>/jest.setup.js"],
 7
 8    // ...
 9    // ...
10
11 };
12
```

Login.jsx

- **Check:**
  - renders login from elements
  - submits form and navigates to /main on successful login
  - calls onLogin with correct user info when login succeeds
  - otherwise, shows alert when login fails

- **Tests** 🧪 **:**
  - ① successful login navigates to /main and calls onLogin
  - ② failed login shows alert

- **Code**
  - ◫ Login.int.test.jsx

- **To Run:**
  - ```
    1 | npm test -- client/src/templates/__tests__/Login.int.test.jsx
    ```

- **All Passed**
  - ✅ successful login navigates to /main and calls onLogin
  - ✅ failed login shows alert

MainTemplate.jsx

- **Check:**
  - renders main layout stably with **Sidebar** and **Dashboard** components
  - successfully fetches and renders both **BaseTemplates** and **CustomTemplates**
  - confirms each section correctly renders the fetched templates
  - and, handles fetch failures (must still render layout)

- **Tests** 🧪 :
  - ❶ renders layout with Sidebar and Dashboard
  - ❷ displays base and custom templates after fetch
  - ❸ handles fetch failure: still renders Sidebar and Dashboard

- **Code**
  - 🔣 MainTemplate.int.test.jsx

- **To Run:**
  - ```
    1  npm test -- client/src/templates/__tests__/MainTemplate.int.test.jsx
    ```

- **All Passed**
  - ✅ renders layout with Sidebar and Dashboard
  - ✅ displays base and custom templates after fetch
  - ✅ handles fetch failure: still renders Sidebar and Dashboard

---

UseScalePage.jsx

- **Check:**
  - renders full layout with **Sidebar**, **HorizontalSidebar**, and **TableSection** components
  - successfully fetches and renders **rows** and **levels** from mocked enpoints

- fetched data are correctly rendered
- handles fetch failures (still renders **Sidebar** and **TableSection)**

- **Tests** 🧪 :
  - 1️⃣ renders layout and fetched data correctly
  - 2️⃣ handles fetch failure: still renders Sidebar and TableSection accordingly

- **Code**
  - 🔣 UseScalePage.int.test.jsx

- **To Run:**
  - ```
    1  npm test -- client/src/templates/__tests__/UseScalePage.int.test.jsx
    ```

- **All Passed**
  - ✅ renders layout and fetched data correctly
  - ✅ handles fetch failure: still renders Sidebar and TableSection accordingly