**Introduction**: Creating an efficient data driven playwright test to login to Asana iterate through JSON test data and verify the expected data is present in the UI.

**Implementation Details**: The script must first login using the given username and password, this is being handled in the beforeEach section that is executed before each test is run. The test cases are represented as an element in the JSON file and include the following attributes: name, app, task, column, tags[]. The tags attribute is an array of tags. The code utilizes loops to iterate through each JSON element and verify the attributes associated with that element. The array of tags associated with that element are also verified in an inner for loop. Adding a new test case would be as simple as adding a new row to the JSON file.

GitHub Repository:

**Challenges and Solutions**:

* Login –In order to make sure the page was loaded I used a locator.waitFor to find if the Logout button was present to verify the login was a success
* Columns – the columns could be easier to locate in the DOM. I used a selector to find the columns and then used filtering to find items under the correct column
* Project names - The projects were documented with a period at the end. These values were updated without the “.” to verify I could properly navigate to the correct project.

**Results**:

* All test cases failed due to the documented project strings in the technical evaluation document. The test cases will pass if the project names are corrected to the actual values found.
  + Test 1 –
    - Navigation fails - The project is documented as "Web Application.", I corrected the project names so that navigation could be verified.
  + Test 2 –
    - Navigation fails - The project is documented as "Web Application.", I corrected the project names so that navigation could be verified.
  + Test 3 –
    - Navigation fails - The project is documented as "Web Application.", I corrected the project names so that navigation could be verified.
  + Test 4 –
    - Navigation fails - The project is documented as "Mobile Application.", I corrected the project names so that navigation could be verified.
  + Test 5 –
    - Navigation fails - The project is documented as "Mobile Application.", I corrected the project names so that navigation could be verified.
  + Test 6 –
    - Navigation fails - The project is documented as "Mobile Application.", I corrected the project names so that navigation could be verified.

**Recommendations**:

* App - Login – Once logged in there is no user displayed and no user settings. I used the Logout to identify if I was logged in.
* App - Column Locator – the columns are difficult to locate, an easier to find locator would be helpful.
* Testing - setup - Login once – instead of logging out and back in for each JSON element, logging in only one time and running all tests would be faster
* Testing - possibilities:
  + Login
    - Different users – Tests could be run for additional users. The JSON elements could include the username and password to use for each test case
    - Invalid user – check error
    - Invalid password – check error
    - Account locked – check error
  + Task
    - owner – verify name
    - date – verify date listed for task
  + Tag
    - Verify color
  + Column
    - The count of items could be verified