

Test Report: test_bjbs_loginFailed

Test Case	Description	Result
-----------	-------------	--------

Test Case Identification

The provided Selenium script contains one test case:

- ****Test Case****: `test_bjbs_login_failed`
- ****Description****: This test case attempts to log in to the BJBS login page with invalid credentials and verifies that the login fails, leaving the user on the same login page.

Test Report

Test Case: `test_bjbs_login_failed`

****Description****: The test navigates to the BJBS login page, enters invalid credentials (username: "Away24", password: "Rusia"), clicks the login button, and verifies that the login fails by ensuring the URL remains the same.

****Expected Results****:

- The login page should load successfully.
- The username and password fields should be located and filled with the specified values.

- The login button should be clicked.
- The URL should not change after attempting to log in with the provided credentials, indicating a failed login attempt.

****Actual Results**:**

1. ****Navigation to the Login Page**:** The script successfully navigates to the URL ``https://silauk.bjbs.co.id/view_login`` and saves a screenshot named ``login_page_loaded.png``.
2. ****Validation of Page Load**:** The script confirms that the current URL matches the expected URL.
3. ****Entering Username**:** The username field is successfully located and the value "Away24" is entered. A screenshot named ``username_entered.png`` is saved.
4. ****Entering Password**:** The password field is successfully located and the value "Rusia" is entered. A screenshot named ``password_entered.png`` is saved.
5. ****Clicking Login Button**:** The login button is successfully located and clicked using JavaScript execution. A screenshot named ``login_button_clicked.png`` is saved.
6. ****Verification of Login Failure**:** The script waits for 5 seconds to see if the page changes or an error occurs. It then checks if the URL remains the same, confirming the login failure.

****Issues Found**:**

- ****Implicit Wait Usage**:** The script uses an implicit wait of 10 seconds, which can lead to unnecessary waiting if elements are found quickly. It is generally better to use explicit waits for specific elements.
- ****Hardcoded Timeout**:** The ``time.sleep(5)`` statement is used for waiting after clicking the login button. This is not a reliable method as the actual time required for the page to respond can vary. An explicit wait should be used instead.
- ****Exception Handling**:** The script includes exception handling for ``TimeoutException``, ``NoSuchElementException``, ``UnexpectedAlertPresentException``, and a generic

`Exception`. However, it could be improved by handling these exceptions more gracefully and providing more detailed error messages.

- **Screenshot Naming**: The screenshots are named based on the step description, which is good practice. However, the screenshot directory is not cleared after the test session, which could lead to clutter over time.

Recommendations:

1. **Use Explicit Waits**: Replace the implicit wait with explicit waits to wait for specific elements to be present or clickable. This will make the script more reliable and faster.

```
```python

WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.CSS_SELECTOR, "input[id='username']")))

```
```

2. **Avoid `time.sleep`**: Use explicit waits to handle dynamic content instead of `time.sleep`.

```
```python

WebDriverWait(driver, 10).until(EC.url_changes(url))

```
```

3. **Enhance Exception Handling**: Improve exception handling by logging detailed error messages and taking additional actions if necessary (e.g., logging the page source for debugging).

```
```python

except TimeoutException as e:

 save_screenshot(driver, 'timeout_exception')
```

```
pytest.fail(f'TimeoutException: {e}. Page source: {driver.page_source}')
```

```
'''
```

4. **\*\*Clear Screenshot Directory\*\***: Implement a cleanup function to delete the screenshot directory after each test session to prevent clutter.

```
```python
```

```
import shutil
```

```
def teardown_module():
```

```
    shutil.rmtree(session_dir)
```

```
'''
```

5. ****Parameterize Test Data****: Consider parameterizing the test data (e.g., username and password) to make the test more flexible and reusable.

By implementing these recommendations, the script will be more robust, maintainable, and efficient.