# Test Case Scenario Report: Login Automation

| Test Case ID | TC\_LOGIN\_001 |
| --- | --- |
| Title | Verify successful login on The‑Internet demo site |
| Objective | Confirm that a user can log in with valid credentials and see a success message. |
| Preconditions | 1. ChromeDriver installed and in PATH 2. Python & Selenium installed 3. Internet connection available |
| Test Data | Username: tomsmith Password: SuperSecretPassword! |
| Environment | macOS (Zsh), Chrome browser, Selenium WebDriver |

## Test Steps

* Launch Chrome via Selenium WebDriver
* Navigate to https://the-internet.herokuapp.com/login
* Wait for the “Username” field to appear (up to 10 s)
* Locate the “Username” input by ID username
* Locate the “Password” input by ID password
* Enter tomsmith into the username field
* Enter SuperSecretPassword! into the password field
* Locate and click the Submit button (button[type='submit'])
* Wait for the success banner (.flash.success) to appear (up to 10 s)
* Print the banner text to the console
* Pause for user inspection (script waits for Enter)
* Close the browser and end session

## Expected Result

After step 8, the page should transition to a secure area. A green banner with text “You logged into a secure area!” should be visible. The console should print that exact message. Browser remains open until the user presses Enter, then closes cleanly.

## Actual Result

Chrome opened and navigated to the login page. Credentials were entered automatically. After clicking “Login,” a green success banner appeared reading “You logged into a secure area!” Console output matched the banner text. Script paused for inspection; upon pressing Enter, browser closed without errors.

## Status

Pass

## Notes & Recommendations

* Enhancement: Parameterize credentials to test invalid or locked‑out users.
* Add’l Check: Verify that the Logout button appears or that the URL changes to /secure.
* Error Handling: Add catch‑blocks for TimeoutException to report when elements never appear.
* Reporting: Integrate into a test framework (e.g., pytest) and generate HTML reports for multiple test cases.