Playwright Python Automation Framework

Release Note
Version: 1.0.0
Date: February 03, 2026
Status: Stable Release

1. Executive Summary

This release introduces a robust, scalable test automation framework built using **Python** and **Playwright**. The framework automate verification of the Swag Labs (SauceDemo) application, implementing the **Page Object Model (POM)** for maintainability and **Pytest** for orchestration.

2. Technology Stack & Key Decisions

Core Technologies

- **Python 3.13+**: Chosen for readability and extensive library support.
- **Playwright (Sync API)**: Provides faster execution than Selenium, native auto-wait capabilities, and robust handling of dynamic content.
- **Pytest**: Industry-standard test runner offering powerful fixture management (dependency injection).

Utilities

- **Pytest-HTML**: Generates self-contained HTML reports with embedded screenshots of failures.

- **Base64**: Used to encode failure screenshots directly into the HTML report string.

3. Module Elaboration

Detailed breakdown of framework components:

`config.py` `Configuration`

Purpose: Centralized configuration management.

Stores global constants (BASE_URL, Timeouts, Credentials) and directory paths to avoid hardcoding values in scripts.

`conftest.py` `Fixtures & Hooks`

Purpose: The backbone of Pytest execution.

Manages browser lifecycle (Session/Page scope fixtures). Crucially, it contains the `pytest_runtest_makereport` hook which captures screenshots on test failures and embeds them into the HTML report.

`pages/base_page.py` `POM Base`

Purpose: Parent class for all page objects.

Wraps raw Playwright methods (click, fill, wait) with error handling. Acts as a safety layer between tests and the raw browser API.

`pages/login_page.py` `Business Logic`

Purpose: Encapsulates Login screen interactions.

Contains specific locators for username/password fields and method `login_to_app(user, pass)`.

`tests/test_saucedemo.py` `Test Scripts`

Purpose: Functional test cases.

Contains the actual assertions and workflows (Login Success, Login Failure, Add to Cart). These scripts are clean and readable, interacting only with Page Objects.

4. Key Features

- **Automatic Failure Analysis**: Screenshots are instantly captured and embedded upon failure.
- **Flakiness Reduction**: Playwright's auto-wait mechanisms reduce timing issues.
- **Clean Architecture**: Strict separation of concerns via Page Object Model.