**Test Strategy Note – SauceDemo Automation**

**Scope:**  
The test automation effort focuses on end-to-end user flows for <https://www.saucedemo.com>, starting from login and continuing through product selection, cart validation, checkout, and order confirmation. The aim is to ensure key functionalities are stable, especially under CI-triggered runs.

**🔍 Key Risks Identified**

We focused on the following **high-risk areas**:

* **Login flow** – Credential handling and session management. This is the gateway; if broken, nothing else is testable.
* **Cart calculations** – We verify that product totals
* **Checkout workflow** – From user info input to final order confirmation, this multi-step form is a prime candidate for regression bugs.
* .

These were chosen based on their **frequency of use**, **impact on user experience**, and **potential for regression**.

**🚀 Automation Prioritization**

We didn’t try to “automate everything.” Instead, we started with what delivers the most value early:

1. **Smoke test** – Quick validation that login, cart, and checkout don’t break.
2. **Critical path** – A user flow covering login → add to cart → checkout → order confirmation.
3. **Negative cases** – Invalid login credentials, missing info on checkout, empty cart submissions.

Items like error messages, visual layout, or edge-case inputs (e.g., XSS) are parked for a later phase unless risk escalates.

**🔄 CI Integration Plan**

* **Trigger:** Tests are designed to run on every **pull request** (PR) into the main branch.This can be achieved using Jenkins or Azure devops pipeline. we can schedule jobs using cron function or during PR merge master branch will run the suite to ensure existing testcases working fine.
* **Parallel execution:** Using Docker containers with the help of selenium grid, tests can be split across browsers to test with isolated browsers.
* In the future, we plan to integrate tagging to group smoke vs. regression vs. extended scenarios for targeted execution using tag concepts in testing suite

**🧰 Tools & Environment**

* Framework: Selenium with TestNG (Java)
* Reporting: Allure and ExtentReports for visibility
* Browsers: Chrome
* Test data: Static user set from SauceDemo (e.g., standard\_user)

**🤝 How Others Can Contribute**

* Clone the repo and install dependencies (README is up-to-date).
* Tests can be triggered with a simple Maven o.
* CI credentials and secrets (if any) are managed via GitHub Actions secrets.