A friendly snapshot: Shalwa Automation Project (SAP) is a Java + Selenium Page Object Model test framework that automates the GUI flows of the SauceDemo site (login, cart, checkout, sidebar, footer links). It's beginner-friendly, modular, and built for clarity and reuse.

Project Title

Shalwa Automation Project (SAP) 🦜 ✨

Brief Description

What it does SAP automates end-to-end GUI scenarios on **sauce demo**: logging in with multiple users, adding/removing items, verifying cart persistence, sorting products, navigating sidebars and social links, and validating checkout flows. The code uses a **Page Object Model** structure (drivers, pages, utilities, tests) for maintainability and readability.

Installation Guide 🚀

Prerequisites

- **Java JDK 11+** installed and JAVA_HOME set.
- **Apache Maven** installed.
- **Chromed river** (or another browser driver) matching your browser version.
- IDE (IntelliJ, Eclipse) recommended.

Step-by-step

1. **Clone the repo**

bash

```bash
git clone https://github.com/your-username/Shalwa_Automation_Project_SAP.git
cd Shalwa_Automation_Project_SAP

```

2. **Set driver path** Edit POMUt.setDriverLocation(...) or set system property before running tests. Example:

java

```java
POMUt.setDriverLocation("src/main/resources/chromedriver.exe");

```

3. **Build**

bash

```bash
mvn clean compile
```

```

```

4. **Run tests**

bash

```
mvn test

```

Or run individual test classes from your IDE.

Usage Instructions with Examples 🧪

How tests are organized

- **driver**: Drivers and DriversOptions — centralizes browser creation.
- **pages**: Page objects (LoginPage, HomePage, CartPage, CheckoutPageone, CheckoutPagetwo, NavigationPage).
- **utility**: helpers (price sorting checks, driver setup).
- **tests**: JUnit tests for each feature (LoginPageTests, HomePageTests, CartPageTests, Checkout tests, Navigation tests).

Example: run a single test class

bash

```
mvn -Dtest=LoginPageTests test

```

Example: programmatic usage

- Create driver: Drivers.getDriver(DriversOptions.CHROME)
- Use page objects:

java

```
LoginPage login = new LoginPage(driver);
HomePage home = login.goToHomePage();
CartPage cart = home.gotoCartPage(driver);

```

Tips

- Use POMUt.setDriverLocation to point to your local driver.
- Tests clear cookies and localStorage after each run to keep state clean.

Contributing 🤝

Want to help? Awesome!

- Fork the repo, create a feature branch, and open a PR.
- Suggested contributions:
    - Add cross-browser CI (GitHub Actions).
    - Improve waits and stability (replace sleeps with explicit waits).
    - Add Page Factory or more granular components.
    - Add reporting (Allure, Surefire reports).
- Keep tests isolated and idempotent. Write clear commit messages and include test evidence (screenshots/logs) when relevant.

License 📜

MIT License — free to use, modify, and distribute. Include attribution and don't hold the author liable. (Add full LICENSE file in repo.)

Closing Note 🦜 💬

Thanks for checking out **Shalwa Automation Project (SAP)** — built with curiosity, coffee, and a chatty parrot named **Shalwa** who insists every test has personality. If you add features, tell Shalwa — she'll squawk approval. Happy testing and welcome to the flock!