

Project Idea :

This project focuses on building a **comprehensive software testing framework** that integrates both **manual** and **automation testing** for different categories of applications: **web-based applications** and **API-based services**. The purpose is to simulate a real-world software testing environment where both front-end and back-end systems are tested thoroughly to ensure quality, reliability, and user satisfaction.

Project description :

1. SauceDemo (Web Application):

SauceDemo is a sample **e-commerce web application** designed specifically for testing purposes. It provides features such as **user login, product catalog, shopping cart, and checkout process**.

In this project, SauceDemo will be used to demonstrate:

- **Manual testing** of functional requirements (e.g., verifying login credentials, adding/removing products from the cart, and completing a purchase).
- **Automation testing** using frameworks like **Selenium / Cypress / Playwright** to cover

regression tests and ensure reliability of the main e-commerce flows.

2. **Restful Booker (API):**

Restful Booker is a publicly available **RESTful API** designed for practicing API testing. It allows operations like **creating, retrieving, updating, and deleting bookings**.

In this project, Restful Booker will be used to demonstrate:

- **Manual API testing** using tools like **Postman**, validating endpoints, response codes, and payload structures.
- **Automation of API testing** using tools such as **Rest Assured / Newman**, ensuring consistency and accuracy of responses across multiple scenarios.

By combining both applications, the project showcases a **real-world testing workflow** that covers:

- **Front-end (UI/Web) testing** through SauceDemo.
- **Back-end (API) testing** through Restful Booker.

3. Combined Testing Workflow (Why Both Applications?):

By working on both SauceDemo (Web) and Restful Booker (API), the project provides **end-to-end coverage**:

- **Front-end testing (UI/Web)**: Focused on user experience, interface functionality, and overall workflow correctness.
- **Back-end testing (API)**: Focused on server-side logic, data validation, and integration of services.

Together, this demonstrates how modern QA engineers operate:

- First validating **visible features** from the user's perspective.
 - Then validating the **underlying services** that support these features.
-

Project Significance:

- Highlights the importance of **both manual and automated testing** in ensuring high-quality applications.
- Prepares a **scalable testing framework** that can be applied to similar projects in real industries.
- Provides practical exposure to **testing tools** widely used in the market (Selenium, Postman, Rest Assured, Cypress, etc.).
- Mimics real-world software testing challenges in **e-commerce systems (UI)** and **booking systems (APIs)**.