# **🧪 Selenium Automation Project Suite**

A comprehensive automation suite built using Selenium WebDriver, Java, TestNG, and Maven, demonstrating real-world automation challenges and best practices. This project captures day-by-day progress through assignments, covering foundational concepts to advanced user interaction and dynamic synchronization techniques.

## **📌 Project Highlights**

* ✅ End-to-End Automation: Login flows, dropdowns, calendar handling, alert popups
* ✅ Real-Time Scenarios: E-commerce cart handling, product validation
* ✅ Wait Strategies: Implicit, Explicit, and Fluent Waits
* ✅ UI Element Interactions: Mouse hover, keyboard control, context click

## **🛠️ Tech Stack**

* Selenium WebDriver
* Java
* TestNG
* Maven

## **🧭 How to Run**

1. Clone the repository
2. Import into your Java IDE (Eclipse/IntelliJ)
3. Use TestNG to run test suites in With TESTNG/
4. Directly run .java files in without TESTNG FreamWork/ for basic execution

## **📚 Learning Journey & Assignments**

### **Day 1: Basic Locators & Login Flow**

* Practiced element locators and page navigation
* Logged in using credentials extracted from UI text

### **Day 2: Static Dropdown**

* Automated dropdown with <select> tag using the Select class

### **Day 3: Dynamic Dropdown**

* Handled dependent dropdowns with dynamic options using XPath

### **Day 4: Suggestive Dropdown**

* Captured autocomplete suggestions using findElements() and loops

### **Day 5: Custom Dropdown**

* Handled passenger count updates using loops and assertions

### **Day 6: Checkbox Testing**

* Verified selection state and counted total checkboxes

### **Assignment 1: Checkbox Exercise**

* Validated default state, toggled selection, and printed checkbox states

### **Day 7: Calendar & UI Element Validation**

* Automated SpiceJet calendar and validated disabled state with CSS properties

### **Assignment 2: UI Form Automation**

* Automated input fields, checkboxes, radio buttons, and dropdowns
* Captured alert and radio label confirmation

### ⚠️ **Alert Handling**

* Handled JavaScript alerts and confirmation popups

### **📦 Tool shop Cart Challenge**

* Selected multiple items from dynamic product list using smart loops and XPath

### **Day 9: Implicit & Explicit Waits**

* Applied global and conditional waits for dynamic loading and alerts

### **Day 10: Fluent Wait**

* Practiced FluentWait on delayed loading content using polling strategy

### **Assignment: Synchronization Strategy**

* Combined implicit and explicit waits for login and cart functionality

### **Day 11: Selenium Actions Class**

* Used Actions class for:
  + Uppercase input
  + Double click
  + Mouse hover
  + Right-click