

Selenium Class 41 - Selenium Project Part-2

Selenium Project

- 1) Project Information
- 2) Project Description
- 3) Stake Holders of the Project
- 4) Interfaces of the Project/AUT
- 5) Features to be tested in Admin Interface
- 6) Features to be tested in User Interface
- 7) Test Requirements, Test scenarios, and Test cases...
- 8) Derive Sanity Test Scenarios for Admin Interface

9) Derive Sanity Test Scenarios for User Interface

- i) Launch the user Interface of the Application
- ii) Verify Required Elements availability of user Interface Homepage...
- iii) Verify Customer Registration with valid data
- iv) Verify Customer Login with valid data
- v) Add product to Shopping cart
- vi) Verify Checkout process
- vii) Verify Customer Track order status etc...

10) Derive Comprehensive (All Possible) Test Scenarios for Admin Interface

- i) Launch Application (includes Launch browser...)
- ii) Verify all elements availability in the Home page of Admin Interface
- iii) Verify Redirect Functionality before Login to the Application (Admin Interface to User Interface)
- iv) Verify Admin Login with valid Username and Valid password
- v) Verify Admin Login with valid Username and invalid password
- vi) Verify Admin Login with invalid Username and valid password
- vii) Verify Admin Login with invalid Username and invalid password
- viii) Verify Admin Login with blank Username and blank password
- ix) Verify Admin Login Locking for 3 failed attempts (Lock for 5 minutes)
- x) Verify Admin Login after 5 minutes with valid data
- xi) Verify all elements availability in the Index page
- xii) Verify Redirect Functionality after Login to the Application (Admin Interface to User Interface)
- xiii) Add Manufacturer
- xiv) Edit Manufacturer
- xv) Delete Manufacturer
- xvi) Add Category
- xvii) Edit Category
- xviii) Delete Category
- xix) Add Product
- xx) Edit Product
- xxi) Delete Product
- xxii) Add Currency
- xxiii) Edit Currency

- xxvi) Delete Currency
- xxv) Set Default Currency
- xxvi) Verify Customer Report/s
- xxvii) Verify product/s Report
- xxviii) Verify Logoff of the Application etc...

11) Derive Comprehensive (All Possible) Test Scenarios for User interface

- i) Verify Launch User Interface with valid url
- ii) Verify all elements availability of user interface Home page
- iii) Verify product "Search"
- iv) Verify "Advanced Search"
- v) Verify Customer Registration with valid data (Positive Test Scenario)
- vi) Verify Customer Registration with invalid data (Negative Test Scenario)
 - (* a) Use invalid Email and date formats
 - b) Use Alphabets for Numeric fields like Phone Number, Post Code etc...
 - c) Use duplicate email id etc...
- vii) Verify Customer Login with valid Email Address and valid Password
- viii) Verify Customer Login with invalid Email Address and valid Password
- ix) Verify Customer Login with valid Email Address and invalid Password
- x) Verify Customer Login with invalid Email Address and inValid Password
- xi) Verify Customer Login with blank Email Address and blank Password
- xii) Verify "Shopping Cart" before selecting any product
- xiii) Verify "Shopping Cart" after selecting any product
- xiv) Verify "Update" in shopping cart while adding a product

- xv) Verify "Update" in shopping cart after adding a product
- xvi) Verify "Remove" the product from shopping cart
- xvii) Checkout before selecting any product and before login
- xviii) Checkout after selecting any product and before login
- xix) Verify Checkout after select any product and after Login
- xx) Verify "Change Address" with another Address
- xxi) Verify availability of all Payment methods
- xxii) Verify Select Payment Method
- xxiii) Verify Checkout process
- xxiv) Verify "Logoff" etc...

Selenium Integration with Maven....

Advantages of Maven Integration:

a) Selenium Test Environment Setup

(You can add all required dependencies like Selenium WebDriver, TestNG etc...to Maven Project (POM xml) then it will download and add those jar file to your project)

b) Maintenance of Selenium Test Environment

(If any new version of project dependencies comes then you just add those dependencies to POM XML then it will download add those jar files...)

c) All of your project team members can get same configuration (by using the POM XML)

(Maintain Common Configuration for all Team Members)

.....

Create Maven Project,

Launch Eclipse IDE

- File Menu
- New
- Project
- Type Maven and Select Maven Project,
- Enter
- Next
- Next
- Enter Group Id
- Enter Artifact Id
- Finish

Then It will create New Maven project

.....

- Add Dependencies from Internet for your Automated Testing in POM (Project Object Model)

- XML file between "Dependencies" tags (Open and Close tag) and Save
Then it will download those dependencies jar files and add to Maven Project

Note: You can edit Dependencies, and share the XML file to your team members, By using this POM XML file, all team members can get same project configuration

Selenium WebDriver-java, TestNG etc...

Create Selenium Test Cases under src/test/java> Java Package>...

.....