# Project Details

ABC Business Solutions has bagged a contract for Automation testing services project with Amazon. As part of the preliminary project acceptance, Amazon has asked ABC Business Solutions to design a POC of Selenium automation project with data-driven framework using standard data exchange files with Maven and TestNG. As you are part the ABC Business Solutions team you are given the task for designing this POC.

# Software Pre-requisites:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Software** | **Version** | | **Download Link** | **Additional Information** |
| **min. reqd.** | **preferred** |
| jdk | 8 or above | 17 or 21 | [Click here](https://www.oracle.com/java/technologies/downloads/) |  |
| Eclipse | 2020-03 | 2020-06 | [Click here](https://www.eclipse.org/downloads/) | IDE used for development |
| Selenium Server | 4.18.1 | 4.18.1 | [Click here](https://github.com/SeleniumHQ/selenium/releases/download/selenium-4.18.0/selenium-server-4.18.1.jar) |  |
| Selenium WebDriver | 4.18.1 | 4.18.1 |  | Add in the pom.xml |
| TestNG | 7.9.0 | 7.9.0 |  |

# Overall Project Details

1. Design a Data Driven project using properties file, xml file, yaml file (if required)
2. Create a Maven Project
3. Create 3 packages
   1. pom 🡺 To store the page object classes and methods
   2. utils 🡺 To perform the common operations
   3. test 🡺 To store the test files
4. Create a Grid Configuration for parallel testing

# Maven Project

1. Create a simple maven project
2. Add the following dependencies
   1. Selenium
   2. TestNG
3. Create a new properties file say config.properties and add the following details
   1. browser=chrome
   2. url=https://www.amazon.in
4. Use utils class to perform the following operations
   1. Read properties file
   2. Setup the browser by reading the properties file method
   3. Getter method to return the driver object
   4. Quit the browser
5. Use test package to perform these operations
   1. Create a BaseTest class
   2. Use BeforeSuite in the BaseTest class to call the
      1. setup browser method
      2. get the driver object
   3. Use AfterSuite to quit the browser
   4. Create rest of test class files and inherit the BaseTest class for the setup and quit browser activities

# Page Object Model

1. Create pom classes with page factory
2. Create constructors in pom classes with parameters to accept the driver object and initialize it along with page factory
3. Create a class for navigation and use this class to navigate to different modules and links
4. Use action class to navigate to Create a Wish List (Accounts and Lists 🡺 Create a Wish List)
5. Create a page for mobile details with reusable methods say,
   1. Validate the total results displayed
   2. Play the mobile video
   3. Select the Avg Customer Review selection
   4. Get the product price
   5. Change the delivery location
   6. Add to cart
   7. Close the “Add to Cart” window
   8. etc…

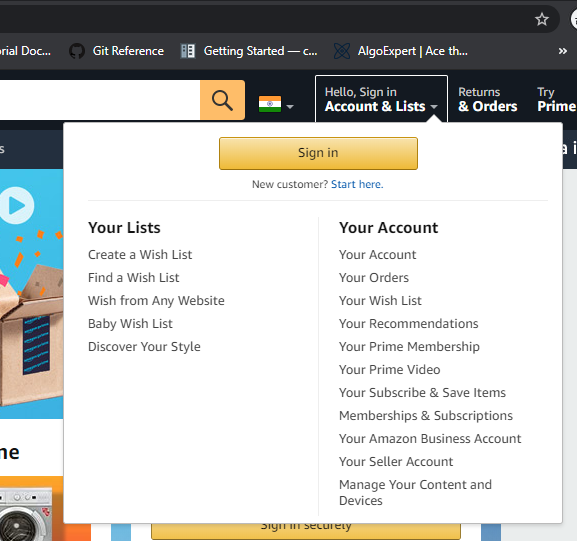
# TestNG

## Test Cases (create as much as details in POM classes and call those methods in the test cases)

## Test Class-1

1. url validation
   1. Enter the url as <http://amzn.in> and check if the url is redirected to <https://www.amazon.in/>
2. Validate the page title “Online Shopping site in India: Shop Online for Mobiles, Books, Watches, Shoes and More - Amazon.in”

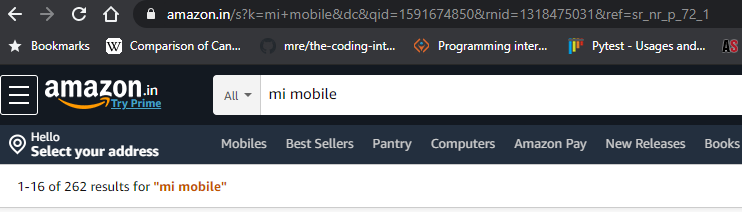
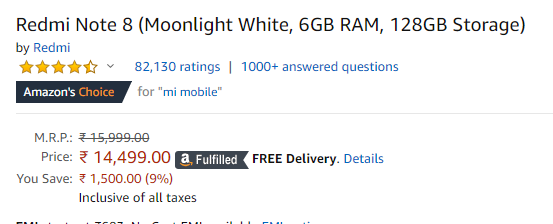
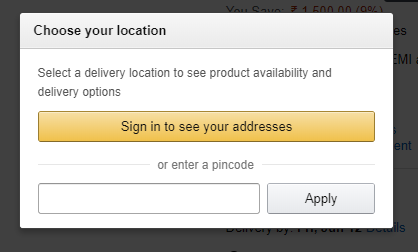
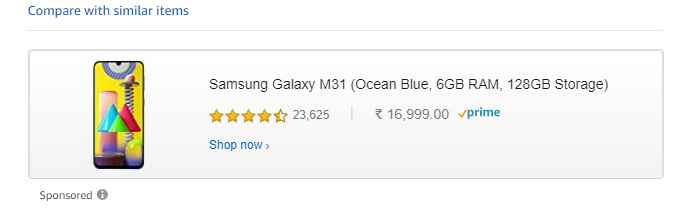
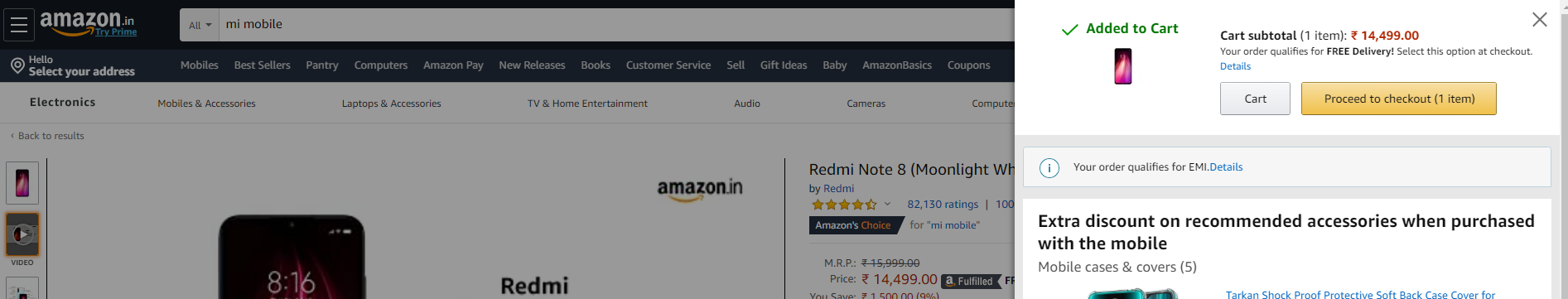
## Test Class-2

1. Use navigation to navigate to Create a Wish List (Accounts and Lists 🡺 Create a Wish List) and verify whether the navigation was successful
2. Use Navigation to navigate to Amazon Pay module and validate the page
3. Use Navigation to navigate New Releases module and validate the page

## Test Class-3

1. Create a testng.xml with parameter as valid and invalid credentials and use them in the test class
2. Create test case for login validation
3. Validate valid login and logout
4. Invalid login validation

## Test Class-4

1. Use navigation to search for a mobile say, “mi mobile”
2. Validate the results, say “1-16 of 264 results for "mi mobile"” 
3. Select “Avg. Customer Review” as 4 start & up” 
4. Select the first mobile say “Redmi Note 8 (Moonlight White, 6GB RAM, 128GB Storage)”
   1. Validate the title is changed to relevant to the mobile select. Say for the above example it should be “Redmi Note 8 (Moonlight White, 6GB RAM, 128GB Storage): Amazon.in: Electronics”
5. Get the mobile price and validate it 
6. Delivery location validation
   1. Click on the “Select delivery location” 
7. Popup validation and pincode validation
   1. wait for the pop up “Choose your location” 
   2. Enter the pincode and click on apply
   3. Validate the change in the delivery location, should display as “Deliver to <city pincode>” as shown below 
8. Click on the sponsored link as shown below 
   1. Use iframe to Click on the link
   2. After clicking the link verify a new window is opened with the mobile details
   3. Navigate to the new window and close it
   4. Navigate back to the previous window
9. Click on add to cart
   1. Wait for the window to be displayed
   2. Validate the “Added to Cart” message
   3. Close the “Added to Cart” window
10. Use Java Script executor to scroll the page down till “Technical Details” is displayed