1. **Sign Up workflow**

Positive cases:

* Test valid user signup with a correct email and strong password.
* Navigate to the website <https://dev.inventory.sumtracker.com/login>
* Verify if ‘Create Account’ is visible for the user
* Click on it
* Verify that all the necessary fields should be present for the user
* User enters First Name, Last Name, Email Address and Password
* Verify if password does not meet all the requirements, then the error should display.
* Click Create Account
* Verify the user is redirected to the Dashboard screen after successful signup.
* Once user successfully logged in then it should navigate to the dashboard

Negative cases:

* Test signup with missing required fields (email, password).
* Navigate to the website <https://dev.inventory.sumtracker.com/login>
* Verify if ‘Create Account’ is visible for the user
* Click on it
* Verify that all the necessary fields should be present for the user
* User enters First Name, Last Name and click on Create Account
* User should be displayed with the error for the missing fields i.e., email and password
* Test signup with an invalid email format.
* Navigate to the website <https://dev.inventory.sumtracker.com/login>
* Verify if ‘Create Account’ is visible for the user
* Click on it
* Verify that all the necessary fields should be present for the user
* User enters First Name, Last Name, Email Address and Password
* User enters email address in invalid format
* Verify user is displayed with ‘Enter valid Email Address’ prompt
* Test signup with a weak password
* Navigate to the website <https://dev.inventory.sumtracker.com/login>
* Verify if ‘Create Account’ is visible for the user
* Click on it
* Verify that all the necessary fields should be present for the user
* User enters First Name, Last Name, Email Address and Password
* User enters password that has some characters same as the First Name
* Verify user is displayed with “The password is too similar to the first name” error
* Test signup with an email already in use.
* Navigate to the website <https://dev.inventory.sumtracker.com/login>
* Verify if ‘Create Account’ is visible for the user
* Click on it
* Verify that all the necessary fields should be present for the user
* User enters First Name, Last Name, Email Address and Password
* User enters Email Address that has already been registered
* Verify user is displayed with the error

1. **Product Creation**

Positive Cases:

* Successfully create a product with valid data (Name, SKU etc).
* Navigate to the link <https://dev.inventory.sumtracker.com/index/products/new>
* User should enter all the required fields for creating the product i.e., Name and SKU
* User can enter additional details
* Click Save
* New product should be created
* Verify the product is listed in the product list after creation.
* Navigate to the link <https://dev.inventory.sumtracker.com/index/products/new>
* User should enter all the required fields for creating the product i.e., Name and SKU
* User can enter additional details
* Click Save
* New product should be created
* Verify that new product should be added in the list after creation

Negative Test Cases:

* Test creating a product with missing required fields (name, SKU).
* Navigate to the link <https://dev.inventory.sumtracker.com/index/products/new>
* User does not enter all the required fields for creating the product i.e., Name and SKU
* User enter additional details
* Click Save
* User is not able to create the new product
* Test creating a product with an invalid SKU format.
* Navigate to the link <https://dev.inventory.sumtracker.com/index/products/new>
* User enters invalid format for SKU field
* User enter other details
* Click Save
* User is not able to create the new product

1. **Inventory Update (Stock by Location)**

Positive Test Cases:

* Successfully update the stock quantity of a product for a specific location.
* Navigate to link <https://dev.inventory.sumtracker.com/index/stock-by-location>
* Search for the product for which stock must be added
* Click on Edit Stock
* Verify user can see three options i.e., Add, Sub, set
* User selects Add option and add the stock
* Stock should be visible in the row

Negative Test Cases:

* Test updating the stock with a negative quantity.
* Navigate to link <https://dev.inventory.sumtracker.com/index/stock-by-location>
* Search for the product for which stock must be added
* Click on Edit Stock
* Verify user can see three options i.e., Add, Sub, set
* User selects Add option and add the stock with the negative quantity
* Stock should not be visible in the row
* Test submitting an empty stock quantity.
* Navigate to link <https://dev.inventory.sumtracker.com/index/stock-by-location>
* Search for the product for which stock must be added
* Click on Edit Stock
* Verify user can see three options i.e., Add, Sub, set
* User selects Add option and add the stock with nothing in the field
* Stock should not be visible in the row

**Automation Tests**

**SignUpPage.Java**

package pages;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

public class SignupPage {

WebDriver driver;

// Locators for elements

By emailField = By.id("email");

By passwordField = By.id("password");

By signupButton = By.id("signup-btn");

By dashboard = By.className("dashboard");

public SignupPage(WebDriver driver) {

this.driver = driver;

}

public void enterSignupDetails(String email, String password) {

driver.findElement(emailField).sendKeys(email);

driver.findElement(passwordField).sendKeys(password);

}

public void clickSignupButton() {

driver.findElement(signupButton).click();

}

public boolean isDashboardVisible() {

try {

WebElement dashboardElement = driver.findElement(dashboard);

return dashboardElement.isDisplayed();

} catch (Exception e) {

return false;

}

}

}

**SignUpTest.Java**

package tests;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.annotations.Test;

import pages.SignupPage;

import utils.ExcelReader;

public class SignupTest {

WebDriver driver;

@Test(dataProvider = "signupData", dataProviderClass = ExcelReader.class)

public void testSignup(String email, String password, boolean expectedResult) {

driver = new ChromeDriver();

driver.get("https://dev.inventory.sumtracker.com/signup");

SignupPage signupPage = new SignupPage(driver);

signupPage.enterSignupDetails(email, password);

signupPage.clickSignupButton();

boolean isDashboardVisible = signupPage.isDashboardVisible();

if (expectedResult) {

Assert.assertTrue(isDashboardVisible, "Signup failed. Dashboard is not visible.");

} else {

Assert.assertFalse(isDashboardVisible, "Signup unexpectedly successful.");

}

driver.quit();

}

}

**ExcelReader.Java  
(Create a excel file with the test data)**

package utils;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import org.testng.annotations.DataProvider;

import java.io.FileInputStream;

import java.io.IOException;

public class ExcelReader {

@DataProvider(name = "signupData")

public static Object[][] getSignupData() throws IOException {

FileInputStream fis = new FileInputStream("src/test/java/data/testData.xlsx");

XSSFWorkbook workbook = new XSSFWorkbook(fis);

Sheet sheet = workbook.getSheetAt(0);

int rowCount = sheet.getPhysicalNumberOfRows();

Object[][] data = new Object[rowCount - 1][3]; // 3 columns

for (int i = 1; i < rowCount; i++) {

Row row = sheet.getRow(i);

data[i - 1][0] = row.getCell(0).getStringCellValue(); // email

data[i - 1][1] = row.getCell(1).getStringCellValue(); // password

data[i - 1][2] = row.getCell(2).getBooleanCellValue(); // expected result (true/false)

}

workbook.close();

return data;

}

}

**ProductTest.java**

package tests;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.annotations.Test;

import pages.ProductPage;

import utils.ExcelReader;

public class ProductTest {

WebDriver driver;

@Test(dataProvider = "productData", dataProviderClass = ExcelReader.class)

public void testProductCreation(String name, String sku, String description, String price, boolean expectedResult) {

driver = new ChromeDriver();

driver.get("https://dev.inventory.sumtracker.com/index/products/new");

ProductPage productPage = new ProductPage(driver);

productPage.enterProductDetails(name, sku, description, price);

productPage.submitProductForm();

boolean isProductCreated = productPage.isProductCreated();

Assert.assertEquals(isProductCreated, expectedResult, "Product creation failed or succeeded unexpectedly.");

driver.quit();

}

}