Create Mini project on Tira Beauty E-Commerce Website

URL= https://www.tirabeauty.com/

1. create module wise requirements and upload in jira tool. in Jira tool create EPIC, Story and Task

2. Create test plan on tira beauty requirement module wise . create in excel sheet.

3. Write test cases for all the possible scenarios(Excel sheet)(At least 10 testcases)

4. Create Automated testcases in TestNG framework using Page object model concept, and add excel file for data driven testing(At least 10 testcases)

5. find out all the defect and create defect report on all the defect, and then upload in jira tool using Bug.

6. Maintain all the testing reports for your positive and negative scenarios

7. Create Test Analysis Report on Tira Beauty project

Module	Testing Types	Test Scope	Responsibility	Priority
User Registration & Login	Functional, Security	Login, Registration, Validation	QA Team	High
Product Browsing & Search	Functional, Usability	Search accuracy, Filters, Sorting	QA Team	High
Shopping Cart & Checkout	Functional, Integration	Add/remove items, Payment processing	QA Team	High
User Profile Management	Functional	Profile updates, Address management	QA Team	Medium
Language & Settings	Functional, Localization	Language change, Localization testing	QA Team	Medium
Order History & Tracking	Functional, Integration	View order history, status tracking	QA Team	Medium

Test Case ID	Module	Test Case Description	Pre-condition	Test Steps	Expected Result
TC001	User Registration	Verify registration with valid input	User on registration page	Enter valid data and submit	Successful registration confirmation
TC002	User Registration	Verify error message for invalid email format	User on registration page	Enter invalid email and submit	Error message for invalid email
TC003	User Login	Verify login with valid credentials	User registered	Enter valid username and password, click login	User is logged in and redirected
TC004	Product Browsing	Verify search results for a product	User logged in	Search for "lipstick"	Relevant products listed with correct titles
TC005	Shopping Cart	Verify adding product to cart	User logged in, product listed	Add product to cart	Product appears in cart with correct details
TC006	Shopping Cart	Verify removing product from the cart	Product exists in cart	Remove product from cart	Cart updated with product removed
TC007	Checkout	Verify successful checkout process	Cart contains product	Proceed to checkout and make payment	Order confirmed with receipt
TC008	User Profile	Verify updating shipping address	User logged in	Edit address details and save	Address updated correctly

Test Case ID	Module	Test Case Description	Pre-condition	Test Steps	Expected Result
TC009	Language Settings	Verify language change functionality	User on any page	Change language setting	Site text updates to selected language
TC010	Order History	Verify order history displays recent orders	User logged in with orders	Navigate to order history	Recent orders displayed with correct details

Sample test cases:

```
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;
import org.testng.Assert;

public class TiraBeautyTests extends BaseTest {

    LoginPage loginPage;
    RegistrationPage registrationPage;
    ProductPage productPage;
    CartPage cartPage;
    CheckoutPage checkoutPage;
    ProfilePage profilePage;
    SettingsPage settingsPage;
    OrderHistoryPage orderHistoryPage;

    @Test(dataProvider = "registrationData")
```

```java
public void testUserRegistration(String email, String password) {
   registrationPage = new RegistrationPage(driver);
   registrationPage.goTo();
   registrationPage.register(email, password);
   Assert.assertTrue(registrationPage.isRegistrationSuccess());
}

@Test(dataProvider = "loginData")
public void testUserLogin(String username, String password) {
   loginPage = new LoginPage(driver);
   loginPage.goTo();
   loginPage.login(username, password);
   Assert.assertTrue(loginPage.isLoggedIn());
}

@Test(dataProvider = "searchProducts")
public void testMultipleProductSearches(String productName) {
   productPage = new ProductPage(driver);
   productPage.search(productName);
   Assert.assertTrue(productPage.isSearchResultDisplayed());
}

@Test(dataProvider = "addToCartData")
public void testAddProductToCart(String productName) {
   productPage = new ProductPage(driver);
   productPage.search(productName);
   productPage.addFirstProductToCart();
   cartPage = new CartPage(driver);
```

```
    Assert.assertTrue(cartPage.isProductInCart(productName));
}

@Test(dataProvider = "removeFromCartData")
public void testRemoveProductFromCart(String productName) {
    cartPage = new CartPage(driver);
    cartPage.removeProduct(productName);
    Assert.assertFalse(cartPage.isProductInCart(productName));
}

@Test(dataProvider = "checkoutData")
public void testCheckoutProcess(String address, String paymentMethod) {
    checkoutPage = new CheckoutPage(driver);
    checkoutPage.proceedToCheckout(address, paymentMethod);
    Assert.assertTrue(checkoutPage.isOrderConfirmed());
}

@Test(dataProvider = "updateProfileData")
public void testUpdateUserProfile(String newName, String newPhone) {
    profilePage = new ProfilePage(driver);
    profilePage.goTo();
    profilePage.updateProfile(newName, newPhone);
    Assert.assertTrue(profilePage.isProfileUpdated(newName));
}

@Test(dataProvider = "changeLanguageData")
public void testChangeLanguage(String language) {
    settingsPage = new SettingsPage(driver);
    settingsPage.changeLanguage(language);
    Assert.assertTrue(settingsPage.isLanguageChanged(language));
}

@Test
public void testShopByCategories() {
    productPage = new ProductPage(driver);
    productPage.goToCategory("Skin Care");
    Assert.assertTrue(productPage.isCategoryPageDisplayed("Skin Care"));
}

@Test
public void testOrderHistoryDisplay() {
    orderHistoryPage = new OrderHistoryPage(driver);
    orderHistoryPage.goTo();
    Assert.assertTrue(orderHistoryPage.isOrderListDisplayed());
}

@DataProvider(name = "registrationData")
public Object[][] getRegistrationData() {
    return ExcelUtils.getData("data/TiraBeautyData.xlsx", "Registration");
}

@DataProvider(name = "loginData")
public Object[][] getLoginData() {
    return ExcelUtils.getData("data/TiraBeautyData.xlsx", "Login");
}
```

```
@DataProvider(name = "searchProducts")
public Object[][] getSearchProducts() {
    return ExcelUtils.getData("data/TiraBeautyData.xlsx", "SearchProducts");
}

@DataProvider(name = "addToCartData")
public Object[][] getAddToCartData() {
    return ExcelUtils.getData("data/TiraBeautyData.xlsx", "AddToCart");
}

@DataProvider(name = "removeFromCartData")
public Object[][] getRemoveFromCartData() {
    return ExcelUtils.getData("data/TiraBeautyData.xlsx", "RemoveFromCart");
}

@DataProvider(name = "checkoutData")
public Object[][] getCheckoutData() {
    return ExcelUtils.getData("data/TiraBeautyData.xlsx", "Checkout");
}

@DataProvider(name = "updateProfileData")
public Object[][] getUpdateProfileData() {
    return ExcelUtils.getData("data/TiraBeautyData.xlsx", "UpdateProfile");
}

@DataProvider(name = "changeLanguageData")
public Object[][] getChangeLanguageData() {
    return ExcelUtils.getData("data/TiraBeautyData.xlsx", "ChangeLanguage");
    }
}
```

Test analysis (example report)

Section	Details
Project Overview	Tira Beauty e-commerce website for makeup, skin, hair, and beauty products
Testing Scope	Functional, integration, UI, usability, security, localization testing
Testing Objectives	Ensure robust user registration, product search, purchase flow, profile management, and settings
Testing Approach	Combination of manual and automated testing, data-driven testing and POM structure in automation
Key Testing Areas	Login/Registration, Product Search, Cart & Checkout, User Profile, Language & Settings, Order History
Test Deliverables	Test plans, test cases, defect reports, automated scripts, test execution & status reports
Defect Summary	Logged critical bugs in login and checkout workflow, medium priority UI discrepancies
Risks & Mitigation	Feature changes during development, 3rd party payment gateway dependency
Recommendations	Increase automation coverage, continuous integration setup, periodic localization review

NOTE : Defect report uploaded in ppt format