Appium Capstone Project Documentation

Name : Aniket Bourasee
Email: aniketbourasee007@gmail.com

Project Title :

E-Commerce App Testing with Appium

Project Overview :

This project involves performing automated testing on an e-commerce app, which is hosted on Sauce Labs. The testing is conducted using Appium, an open-source automation tool for mobile applications. The goal of this project is to ensure the functionality, performance, and usability of these applications on Android devices through various test scenarios.

Technology Stack :

- **Appium**: For automated mobile testing
- **Java**: For scripting the test cases
- **JUnit**: For test case management and execution
- **Sauce Labs**: For running tests on cloud-based emulator or real devices
- **Android Emulator**: For local testing
- **UiAutomator2**: Automation engine for Android
- **Maven**: For dependency management
- **Appium Inspector** : Use to inspect the elements of app

Project Components

1. Test Automation Environment :

- **Appium Version**: v2.12.1
- **Android Emulator Version**: Android 8.1
- **Device Information**:
    - Device Name: Pixel_2_API_27
    - OS Version: Android 8.1

2. Desired Capabilities :

These capabilities are used to configure the Appium test session and define the parameters for the app under test.

```
capablities.setCapability("appium:deviceName", "Pixel_2_API_27");
capablities.setCapability("appium:platformName", "Android");
capablities.setCapability("appium:automationName", "UiAutomator2");
capablities.setCapability("appium:platformVersion", "8.1");
capablities.setCapability("appium:appPackage", "com.swaglabsmobileapp");
capablities.setCapability("appium:appActivity", "com.swaglabsmobileapp.MainActivity");
capablities.setCapability("appium:newCommandTimeout", 120);
```

3. Test Scenarios :

The project includes test cases for both the food delivery and e-commerce applications, such as:

- **Login/Signup Verification**: Testing user login and registration functionality.
- **Product Search and Checkout**: Verifying product search, selection, and checkout functionality.
- **UI Consistency**: Ensuring the UI elements like buttons, text fields, and menus work as expected.
- **Performance Testing**: Checking app performance under normal and stress conditions.
- **Error Handling**: Ensuring the app handles errors (e.g., wrong inputs, no internet) gracefully.

4. Test Execution :

Tests are executed on Android devices via Appium, using local Android emulators for broader test coverage.

5. Reporting :

Test results are captured through **Aventstack ExtentReports** and are integrated into the Appium framework for detailed reporting. ExtentReports provides rich HTML reports that can be easily customized to include logs, test results, screenshots, and execution information.

6. **Project Dependencies :**

- **TestNG (7.9.0)**
  A testing framework used to manage and run test cases with support for parallel execution and configuration.
- **Appium Java Client (9.3.0)**
  Library for interacting with Appium to automate mobile app testing on Android and iOS.
- **ExtentReports (5.0.9)**
  Tool for generating HTML reports with logs, screenshots, and test status for better analysis and debugging.

7. Project Demo :

https://drive.google.com/file/d/1pzyHXNJU2uKjmmc4XjjSqgmqneNQ4xpg/view?usp=sharing

Steps :

1. Create a new Maven project on Eclipse IDE and add the following dependencies:

```xml
<dependencies>
 <dependency>
  <groupId>org.testng</groupId>
  <artifactId>testng</artifactId>
  <version>7.9.0</version>
  <scope>test</scope>
 </dependency>
 <!-- Appium Java Client -->
 <dependency>
  <groupId>io.appium</groupId>
  <artifactId>java-client</artifactId>
  <version>9.3.0</version>
 </dependency>
 <!-- ExtentReports for reporting (optional) -->
 <dependency>
  <groupId>com.aventstack</groupId>
  <artifactId>extentreports</artifactId>
  <version>5.0.9</version>
 </dependency>
</dependencies>
```

2. Create a **Baseclass** under “src/test/java”.

TestCases.java | SwitchHandeling.java | W3CHttpResponseCodec.class | Baseclass.java | AppiumCaptsone/pom.xml

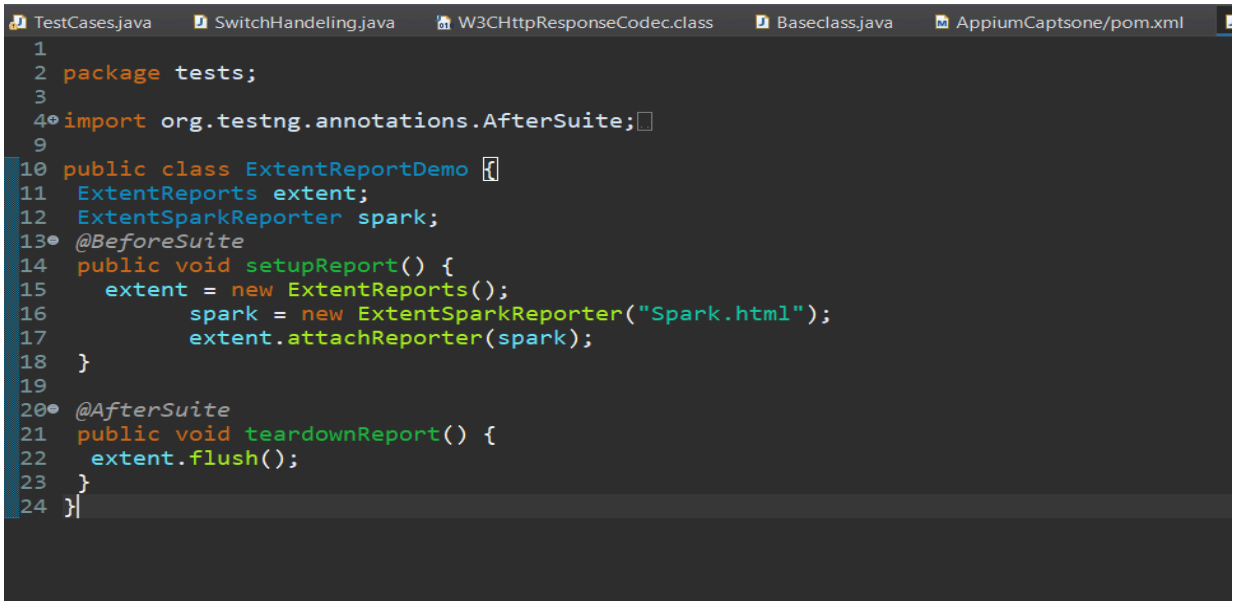
```java
package tests;

import java.net.MalformedURLException;

public class Baseclass extends ExtentReportDemo{
AndroidDriver driver;
@BeforeTest
 public void setup() throws MalformedURLException {
  DesiredCapabilities capablities = new DesiredCapabilities();
  capablities.setCapability("appium:deviceName", "Pixel_2_API_27");
  capablities.setCapability("appium:platformName", "Android");
  capablities.setCapability("appium:automationName", "UiAutomator2");
  capablities.setCapability("appium:platformVersion", "8.1");
  capablities.setCapability("appium:appPackage", "com.swaglabsmobileapp");
  capablities.setCapability("appium:appActivity", "com.swaglabsmobileapp.MainActivity");
  capablities.setCapability("appium:newCommandTimeout", 120);

  URL url = URI.create("http://127.0.0.1:4723/").toURL();
  driver = new AndroidDriver(url, capablities);
 }

@AfterTest
 public void tearDown() {
  driver.quit();

 }
}
```

The **Baseclass** is a test setup and teardown class used for initializing and closing the Appium driver in Android automation tests.

- Setup (@BeforeTest): It configures the Appium driver with desired capabilities, such as the device name, platform, automation engine, app package, and app activity, before running the tests.
- Teardown (@AfterTest): It ensures the Appium driver is properly closed after the tests are completed.

This class serves as the foundation for the tests, allowing for a consistent and reusable test environment setup.

3. Create another class that is “ExtentReportDemo”.



The ExtentReportDemo class is responsible for setting up and generating ExtentReports for the test suite.

- **Setup (@BeforeSuite)**: It initializes the ExtentReports object and the ExtentSparkReporter, which generates HTML reports. The reporter is then attached to the ExtentReports instance to record test results.
- **Teardown (@AfterSuite)**: After the test suite execution, it calls flush() to ensure that the final test results are saved into the report.

4. Create a Class which is “TestCases”.

- **TestCase 1:**

```java
@Test(priority = 1)
Run | Debug
public void testLogin() throws InterruptedException {
    System.out.println("Starting execution of testLogin.");
    ExtentTest loginTest = extent.createTest("LoginTest", "Testing the Login Functionality");
    loginTest.log(Status.INFO, "Login Test has Started");

    // Wait for the login screen to appear
    Thread.sleep(3000);
    System.out.println("Login screen appeared. Waiting for 3 seconds.");

    // Click the "standard_user" button
    WebElement standardUserButton = driver.findElement(AppiumBy.androidUIAutomator(
            "new UiScrollable(new UiSelector().scrollable(true)).scrollIntoView(new UiSelector().text(\"standard_user\"))"));
    standardUserButton.click();
    System.out.println("Clicked the 'standard_user' button.");

    // Scroll up after clicking the button
    WebElement usernamefield = driver.findElement(AppiumBy.androidUIAutomator(
            "new UiScrollable(new UiSelector().scrollable(true)).scrollIntoView(new UiSelector().description(\"test-Username\"))"));
    System.out.println("Scrolled to the 'Username' field.");

    // Wait for 2 seconds after scrolling up
    Thread.sleep(2000);
    System.out.println("Waited for 2 seconds after scrolling.");

    // Click the login button after the 2-second wait
    WebElement loginButton = driver.findElement(By.xpath("//android.view.ViewGroup[@content-desc='test-LOGIN']"));
    loginButton.click();
    System.out.println("Clicked the 'LOGIN' button.");

    // Wait for the product screen to appear
    Thread.sleep(3000);
    System.out.println("Product screen appeared. Waiting for 3 seconds.");

    System.out.println("testLogin execution completed.");
    // Do not close the app here, leave the session open for the next test
}
```

Steps:

- Waits for the login screen to appear.
- Click the "standard_user" button on the login screen.
- Scrolls to the "Username" field.
- Waits for 2 seconds after scrolling.
- Click the "LOGIN" button.
- Waits for the product screen to appear.

- **TestCase 2:**

```java
@Test(priority = 2, dependsOnMethods = "testLogin")
Run | Debug
public void testMenuInteraction() throws InterruptedException {
    System.out.println("Starting execution of testMenuInteraction.");
    ExtentTest menuTest = extent.createTest("MenuTest", "Testing Menu Interaction");
    menuTest.log(Status.INFO, "Menu Interaction Test has Started");

    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(20));

    // Wait for and click the Menu button
    System.out.println("Waiting for Menu button to become visible.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.view.ViewGroup[@content-desc='test-Menu']/android.view.ViewGroup/android.widget.ImageView")));
    System.out.println("Menu button is visible. Clicking on it.");
    driver.findElement(By.xpath("//android.view.ViewGroup[@content-desc='test-Menu']/android.view.ViewGroup/android.widget.ImageView")).click();
    menuTest.log(Status.INFO, "Menu button clicked.");
    System.out.println("Menu button clicked.");

    // Wait for 5 seconds (optional, if you want to add a wait before interacting with the menu)
    System.out.println("Waiting for 5 seconds before interacting with the menu.");
    Thread.sleep(5000);
    System.out.println("Wait time completed.");

    // Click on the first menu option: "ALL ITEMS"
    System.out.println("Waiting for 'ALL ITEMS' menu option to become visible.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.widget.TextView[@text='ALL ITEMS']")));
    System.out.println("'ALL ITEMS' menu option is visible. Clicking on it.");
    driver.findElement(By.xpath("//android.widget.TextView[@text='ALL ITEMS']")).click();
    menuTest.log(Status.PASS, "First menu item clicked: ALL ITEMS");
    System.out.println("'ALL ITEMS' menu option clicked.");

    // Wait for 3 seconds after navigating to the "ALL ITEMS" screen
    System.out.println("Waiting for 3 seconds after navigating to 'ALL ITEMS' screen.");
    Thread.sleep(3000);
    System.out.println("Wait time completed.");

    // Indicate the end of the test execution
    System.out.println("testMenuInteraction execution completed.");
    // Optionally, you could uncomment the following lines to close the app:
    // driver.quit();
    // System.out.println("App closed after 3 seconds.");
}
```

Steps:

- Waits for the menu button to become visible and clicks it.
- Wait for 5 seconds before interacting with the menu.
- Waits for the "ALL ITEMS" menu option to appear and clicks it.
- Waits for 3 seconds after navigating to the "ALL ITEMS" screen.

- **TestCase 3:**

```java
@Test(priority = 3, dependsOnMethods = "testLogin")
Run | Debug
public void testProductInteraction() throws InterruptedException {
    System.out.println("Starting execution of testProductInteraction.");
    ExtentTest menuTest = extent.createTest("MenuTest", "Testing Menu Interaction");
    menuTest.log(Status.INFO, "Menu Interaction Test has Started");

    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(20));

    // Skipping the menu interaction as it's assumed to be handled already.
    System.out.println("Assuming menu interaction is already handled.");

    // Scroll Down Slowly using UiScrollable
    System.out.println("Starting to scroll down slowly.");
    for (int i = 0; i < 5; i++) {
        driver.findElement(AppiumBy.androidUIAutomator(
            "new UiScrollable(new UiSelector().scrollable(true)).scrollForward()"));
        Thread.sleep(1000);  // Slow down the scroll
        System.out.println("Scrolling down step " + (i + 1));
    }
    menuTest.log(Status.INFO, "Scrolled down slowly.");
    System.out.println("Completed scrolling down slowly.");

    // Scroll Up Slowly using UiScrollable
    System.out.println("Starting to scroll up slowly.");
    for (int i = 0; i < 5; i++) {
        driver.findElement(AppiumBy.androidUIAutomator(
            "new UiScrollable(new UiSelector().scrollable(true)).scrollBackward()"));
        Thread.sleep(1000);  // Slow down the scroll
        System.out.println("Scrolling up step " + (i + 1));
    }
    menuTest.log(Status.INFO, "Scrolled up slowly.");
    System.out.println("Completed scrolling up slowly.");

    // Wait for the "View" button and click it (First Click)
    System.out.println("Waiting for 'View' button to become visible.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.view.ViewGroup[@content-desc='test-Toggle']/android.widget.ImageView")));
    System.out.println("'View' button is visible. Clicking on it.");
    driver.findElement(By.xpath("//android.view.ViewGroup[@content-desc='test-Toggle']/android.widget.ImageView")).click();
    menuTest.log(Status.INFO, "Clicked 'View' button to toggle layout.");
    System.out.println("Clicked 'View' button.");

    // Wait for layout to refresh after clicking the "View" button
    // Scroll Down until the third product row (First "View" Click)
    System.out.println("Scrolling down to the third product row.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.widget.ScrollView[@content-desc='test-PRODUCTS']/android.view.ViewGroup/android.view.ViewGroup[3]")));
    driver.findElement(AppiumBy.androidUIAutomator(
        "new UiScrollable(new UiSelector().scrollable(true)).scrollForward()"));
    menuTest.log(Status.INFO, "Scrolled down to third product row after first 'View' click.");
    System.out.println("Scrolled down to the third product row.");

    // Scroll Up after reaching the third product row
    System.out.println("Starting to scroll up after reaching the third product row.");
    for (int i = 0; i < 5; i++) {
        driver.findElement(AppiumBy.androidUIAutomator(
            "new UiScrollable(new UiSelector().scrollable(true)).scrollBackward()"));
        Thread.sleep(1000);  // Slow down the scroll
        System.out.println("Scrolling up step " + (i + 1));
    }
    menuTest.log(Status.INFO, "Scrolled up after reaching third product row.");
    System.out.println("Completed scrolling up after reaching the third product row.");

    // Wait for 2 seconds after scrolling up
    System.out.println("Waiting for 2 seconds after scrolling up.");
    Thread.sleep(2000);
    System.out.println("Wait time completed.");

    // Indicate the end of the test execution
    System.out.println("testProductInteraction execution completed.");
    menuTest.log(Status.PASS, "Test completed successfully after menu interaction.");
}
```

Steps:

1. Assumes the menu interaction is handled from a previous test.
2. Scrolls down slowly through the product list.
3. Scrolls back up after reaching the bottom of the list.
4. Waits for the "View" button to become visible and clicks it to toggle the layout.
5. Waits for the UI to refresh after clicking the "View" button.
6. Scrolls down to the third product row.
7. Scrolls back up after reaching the third product row.

- **TestCase 4:**

```java
@Test(priority = 4, dependsOnMethods = "testLogin")
Run | Debug
public void testAddToCart() throws InterruptedException {
    System.out.println("Starting execution of testAddToCart.");
    ExtentTest addToCartTest = extent.createTest("AddToCartTest", "Testing Add to Cart Interaction");
    addToCartTest.log(Status.INFO, "Add to Cart Test has Started");

    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(30)); // Set the wait time to 30 seconds

    // 1. Click on the first product
    System.out.println("Waiting for the first product to become visible.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("(//android.view.ViewGroup[@content-desc='test-Item'])[1]")));
    System.out.println("First product is visible. Clicking on it.");
    driver.findElement(By.xpath("(//android.view.ViewGroup[@content-desc='test-Item'])[1]")).click();
    addToCartTest.log(Status.INFO, "Clicked the first product.");
    System.out.println("Clicked the first product.");

    // 2. Scroll down to the Inventory item page
    System.out.println("Waiting for the Inventory item page to become visible.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.view.ViewGroup[@content-desc='test-Inventory item page']")));
    System.out.println("Inventory item page is visible. Scrolling down.");
    driver.findElement(AppiumBy.androidUIAutomator(
        "new UiScrollable(new UiSelector().scrollable(true)).scrollForward()"));
    Thread.sleep(1000);  // Slow down the scroll
    addToCartTest.log(Status.INFO, "Scrolled down to the Inventory item page.");
    System.out.println("Scrolled down to the Inventory item page.");

    // 3. Wait for 2 seconds before clicking the "Add to Cart" button
    System.out.println("Waiting for 2 seconds before clicking the 'Add to Cart' button.");
    Thread.sleep(2000);
    System.out.println("Wait time completed.");

    // 4. Click on the "Add to Cart" button
    System.out.println("Waiting for the 'Add to Cart' button to become visible.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.view.ViewGroup[@content-desc='test-ADD TO CART']")));
    System.out.println("'Add to Cart' button is visible. Clicking on it.");
    driver.findElement(By.xpath("//android.view.ViewGroup[@content-desc='test-ADD TO CART']")).click();
    addToCartTest.log(Status.INFO, "Clicked the 'Add to Cart' button.");
    System.out.println("Clicked the 'Add to Cart' button.");

    // 5. Wait for 2 seconds after clicking the "Add to Cart" button
    System.out.println("Waiting for 2 seconds after clicking the 'Add to Cart' button.");
    Thread.sleep(2000);
    System.out.println("Wait time completed.");

    // 6. Click on the Back button to return to the products page
    System.out.println("Waiting for the 'Back to Products' button to become visible.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.view.ViewGroup[@content-desc='test-BACK TO PRODUCTS']")));
    System.out.println("'Back to Products' button is visible. Clicking on it.");
    driver.findElement(By.xpath("//android.view.ViewGroup[@content-desc='test-BACK TO PRODUCTS']")).click();
    addToCartTest.log(Status.INFO, "Clicked the 'Back to Products' button.");
    System.out.println("Clicked the 'Back to Products' button.");

    // 7. Wait for 3 seconds after clicking the Back button
    System.out.println("Waiting for 3 seconds after clicking the 'Back to Products' button.");
    Thread.sleep(3000);
    System.out.println("Wait time completed.");

    addToCartTest.log(Status.PASS, "Add to Cart Test completed successfully.");
    System.out.println("testAddToCart execution completed.");
}
```

Steps:

1. Waits for the first product to become visible and clicks on it.
2. Scrolls down to the Inventory item page.
3. Wait for 2 seconds before clicking the "Add to Cart" button.
4. Click the "Add to Cart" button.
5. Waits for 2 seconds after clicking the "Add to Cart" button.
6. Click the "Back to Products" button to return to the product page.
7. Waits for 3 seconds after clicking the "Back to Products" button.

- **TestCase 5:**

```java
@Test(priority = 5, dependsOnMethods = "testLogin")
Run | Debug
public void testCheckoutProcess() throws InterruptedException {
    System.out.println("Starting execution of testCheckoutProcess.");
    ExtentTest checkoutTest = extent.createTest("CheckoutTest", "Testing the Checkout Process");
    checkoutTest.log(Status.INFO, "Checkout Test has Started");

    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(30)); // Set the wait time to 30 seconds

    // 1. Click on the Cart button
    System.out.println("Waiting for the Cart button to become visible.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.view.ViewGroup[@content-desc='test-Cart']/android.view.ViewGroup/android.widget.ImageView")));
    System.out.println("Cart button is visible. Clicking on it.");
    driver.findElement(By.xpath("//android.view.ViewGroup[@content-desc='test-Cart']/android.view.ViewGroup/android.widget.ImageView")).click();
    checkoutTest.log(Status.INFO, "Clicked the Cart button.");
    System.out.println("Clicked the Cart button.");

    // 2. Wait for 1 second and then scroll down to pause the scroll at the given position
    System.out.println("Waiting for 1 second before scrolling.");
    Thread.sleep(1000);  // Wait for 1 second
    System.out.println("Scrolling down and pausing at the specified position.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.widget.ScrollView[@content-desc='test-Cart Content']/android.view.ViewGroup/android.view.ViewGroup[2]")));
    driver.findElement(AppiumBy.androidUIAutomator(
        "new UiScrollable(new UiSelector().scrollable(true)).scrollForward()"));
    Thread.sleep(1000);  // Slow down the scroll and pause it at the specified position
    checkoutTest.log(Status.INFO, "Scrolled down and paused at the specified position.");

    // 3. Click on the Checkout button
    System.out.println("Waiting for the 'Checkout' button to become visible.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.view.ViewGroup[@content-desc='test-CHECKOUT']")));
    System.out.println("'Checkout' button is visible. Clicking on it.");
    driver.findElement(By.xpath("//android.view.ViewGroup[@content-desc='test-CHECKOUT']")).click();
    checkoutTest.log(Status.INFO, "Clicked the 'Checkout' button.");
    System.out.println("Clicked the 'Checkout' button.");

    // 4. Wait for 4 seconds
    System.out.println("Waiting for 4 seconds after clicking the 'Checkout' button.");
    Thread.sleep(4000);  // Wait for 4 seconds

    checkoutTest.log(Status.PASS, "Checkout Test completed successfully.");
    System.out.println("Completed execution of testCheckoutProcess.");
}
```

Steps:

- Click the Cart button: Waits for the Cart button to become visible and clicks on it.
- Scroll down and pause: Waits for 1 second, then scrolls down to a specified position and pauses.
- Click the Checkout button: Waits for the Checkout button to become visible and clicks on it.
- Wait for 4 seconds after clicking the Checkout button.

- **TestCase 6:**

```java
@Test(priority = 6, dependsOnMethods = "testCheckoutProcess")
Run | Debug
public void testFillCustomerDetails() throws InterruptedException {
    System.out.println("Starting execution of testFillCustomerDetails.");
    ExtentTest customerDetailsTest = extent.createTest("CustomerDetailsTest", "Filling the customer details");
    customerDetailsTest.log(Status.INFO, "Customer Details Test has Started");

    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(30)); // Set the wait time to 30 seconds

    // 1. Fill in the First Name field
    System.out.println("Waiting for the First Name field to become visible.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.widget.EditText[@content-desc='test-First Name']")));
    System.out.println("First Name field is visible. Entering 'Aniket'.");
    driver.findElement(By.xpath("//android.widget.EditText[@content-desc='test-First Name']")).sendKeys("Aniket");
    customerDetailsTest.log(Status.INFO, "Entered 'Aniket' in the First Name field.");
    System.out.println("Entered 'Aniket' in the First Name field.");

    // 2. Fill in the Last Name field
    System.out.println("Waiting for the Last Name field to become visible.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.widget.EditText[@content-desc='test-Last Name']")));
    System.out.println("Last Name field is visible. Entering 'Bourasee'.");
    driver.findElement(By.xpath("//android.widget.EditText[@content-desc='test-Last Name']")).sendKeys("Bourasee");
    customerDetailsTest.log(Status.INFO, "Entered 'Bourasee' in the Last Name field.");
    System.out.println("Entered 'Bourasee' in the Last Name field.");

    // 3. Fill in the Zip/Postal Code field
    System.out.println("Waiting for the Zip/Postal Code field to become visible.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.widget.EditText[@content-desc='test-Zip/Postal Code']")));
    System.out.println("Zip/Postal Code field is visible. Entering '123456'.");
    driver.findElement(By.xpath("//android.widget.EditText[@content-desc='test-Zip/Postal Code']")).sendKeys("123456");
    customerDetailsTest.log(Status.INFO, "Entered '123456' in the Zip/Postal Code field.");
    System.out.println("Entered '123456' in the Zip/Postal Code field.");

    // 4. Click on the Continue button
    System.out.println("Waiting for the 'Continue' button to become visible.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.view.ViewGroup[@content-desc='test-CONTINUE']")));
    System.out.println("'Continue' button is visible. Clicking on it.");
    driver.findElement(By.xpath("//android.view.ViewGroup[@content-desc='test-CONTINUE']")).click();
    customerDetailsTest.log(Status.INFO, "Clicked the 'Continue' button.");
    System.out.println("Clicked the 'Continue' button.");

    // 5. Wait for 3 seconds after clicking the Continue button
    System.out.println("Waiting for 3 seconds after clicking the 'Continue' button.");
    Thread.sleep(3000);  // Wait for 3 seconds

    customerDetailsTest.log(Status.PASS, "Customer Details Test completed successfully.");
    System.out.println("Completed execution of testFillCustomerDetails.");
}
```

Steps:

- Fill in the First Name field: Waits for the First Name field to become visible and enters the value "Aniket."
- Fill in the Last Name field: Waits for the Last Name field to become visible and enters the value "Bourasee."
- Fill in the Zip/Postal Code field: Waits for the Zip/Postal Code field to become visible and enters the value "123456."
- Click on the Continue button: Waits for the Continue button to become visible and clicks on it.
- Wait for 3 seconds: Wait for 3 seconds after clicking the Continue button.

- **TestCase 7:**

```java
@Test(priority = 7, dependsOnMethods = "testFillCustomerDetails")
Run | Debug
public void testFinishCheckout() throws InterruptedException {
    System.out.println("Starting execution of testFinishCheckout.");
    ExtentTest finishCheckoutTest = extent.createTest("FinishCheckoutTest", "Finalizing the Checkout Process");
    finishCheckoutTest.log(Status.INFO, "Finish Checkout Test has Started");

    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(30)); // Set the wait time to 30 seconds

    // 1. Slowly scroll down and pause the scroll at the checkout overview button
    System.out.println("Waiting for 1 second before scrolling.");
    Thread.sleep(1000);  // Wait for 1 second

    System.out.println("Scrolling down and pausing at the checkout overview section.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.widget.ScrollView[@content-desc='test-CHECKOUT: OVERVIEW']/android.view.ViewGroup/android.view.ViewGroup[2]")));
    driver.findElement(AppiumBy.androidUIAutomator(
        "new UiScrollable(new UiSelector().scrollable(true)).scrollForward()"));
    Thread.sleep(1000);  // Slow down the scroll and pause it at the specified position
    finishCheckoutTest.log(Status.INFO, "Scrolled down and paused at the specified position.");
    System.out.println("Scrolled down and paused at the specified position.");

    // 2. Click on the Finish button
    System.out.println("Waiting for the 'Finish' button to become visible.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.view.ViewGroup[@content-desc='test-FINISH']")));
    System.out.println("'Finish' button is visible. Clicking on it.");
    driver.findElement(By.xpath("//android.view.ViewGroup[@content-desc='test-FINISH']")).click();
    finishCheckoutTest.log(Status.INFO, "Clicked the 'Finish' button.");
    System.out.println("Clicked the 'Finish' button.");

    // 3. Wait for 4 seconds after clicking the Finish button
    System.out.println("Waiting for 4 seconds after clicking the 'Finish' button.");
    Thread.sleep(4000);

    finishCheckoutTest.log(Status.PASS, "Finish Checkout Test completed successfully.");
    System.out.println("Completed execution of testFinishCheckout.");
}
```

Steps:

- Scroll down and pause at the checkout overview section: Waits for 1 second before starting the scroll and then scrolls down to the checkout overview section, pausing the scroll.
- Click on the Finish button: Waits for the Finish button to become visible and clicks on it.
- Wait for 4 seconds: Waits for 4 seconds after clicking the Finish button to allow the checkout completion process.

- **TestCase 8:**

```
@Test(priority = 8, dependsOnMethods = "testFinishCheckout")
Run | Debug
public void testLogout() throws InterruptedException {
    System.out.println("Starting execution of testLogout.");
    ExtentTest logoutTest = extent.createTest("LogoutTest", "Testing Logout Functionality");
    logoutTest.log(Status.INFO, "Logout Test has Started");

    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(30)); // Set the wait time to 30 seconds

    // 1. Wait for 2 seconds then click on the 'BACK HOME' button
    System.out.println("Waiting for 2 seconds before clicking the 'BACK HOME' button.");
    Thread.sleep(2000);  // Wait for 2 seconds
    System.out.println("Clicking the 'BACK HOME' button.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.view.ViewGroup[@content-desc='test-BACK HOME']")));
    driver.findElement(By.xpath("//android.view.ViewGroup[@content-desc='test-BACK HOME']")).click();
    logoutTest.log(Status.INFO, "Clicked on the 'BACK HOME' button.");
    System.out.println("'BACK HOME' button clicked.");

    // 2. Click on the 'Menu' button
    System.out.println("Waiting for the 'Menu' button to become visible.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.view.ViewGroup[@content-desc='test-Menu']/android.view.ViewGroup/android.widget.ImageView")));
    System.out.println("'Menu' button is visible. Clicking on it.");
    driver.findElement(By.xpath("//android.view.ViewGroup[@content-desc='test-Menu']/android.view.ViewGroup/android.widget.ImageView")).click();
    logoutTest.log(Status.INFO, "Clicked on the 'Menu' button.");
    System.out.println("'Menu' button clicked.");

    // 3. Wait for 2 seconds and click on the 'LOGOUT' button
    System.out.println("Waiting for 2 seconds before clicking the 'LOGOUT' button.");
    Thread.sleep(2000);  // Wait for 2 seconds
    System.out.println("Clicking the 'LOGOUT' button.");
    wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.widget.TextView[@text='LOGOUT']")));
    driver.findElement(By.xpath("//android.widget.TextView[@text='LOGOUT']")).click();
    logoutTest.log(Status.INFO, "Clicked on the 'LOGOUT' button.");
    System.out.println("'LOGOUT' button clicked.");

    // 4. Wait for 5 seconds after clicking 'LOGOUT'
    System.out.println("Waiting for 5 seconds after clicking the 'LOGOUT' button.");
    Thread.sleep(5000);  // Wait for 5 seconds

    logoutTest.log(Status.PASS, "Logout Test completed successfully.");
    System.out.println("Completed execution of testLogout.");
}
```

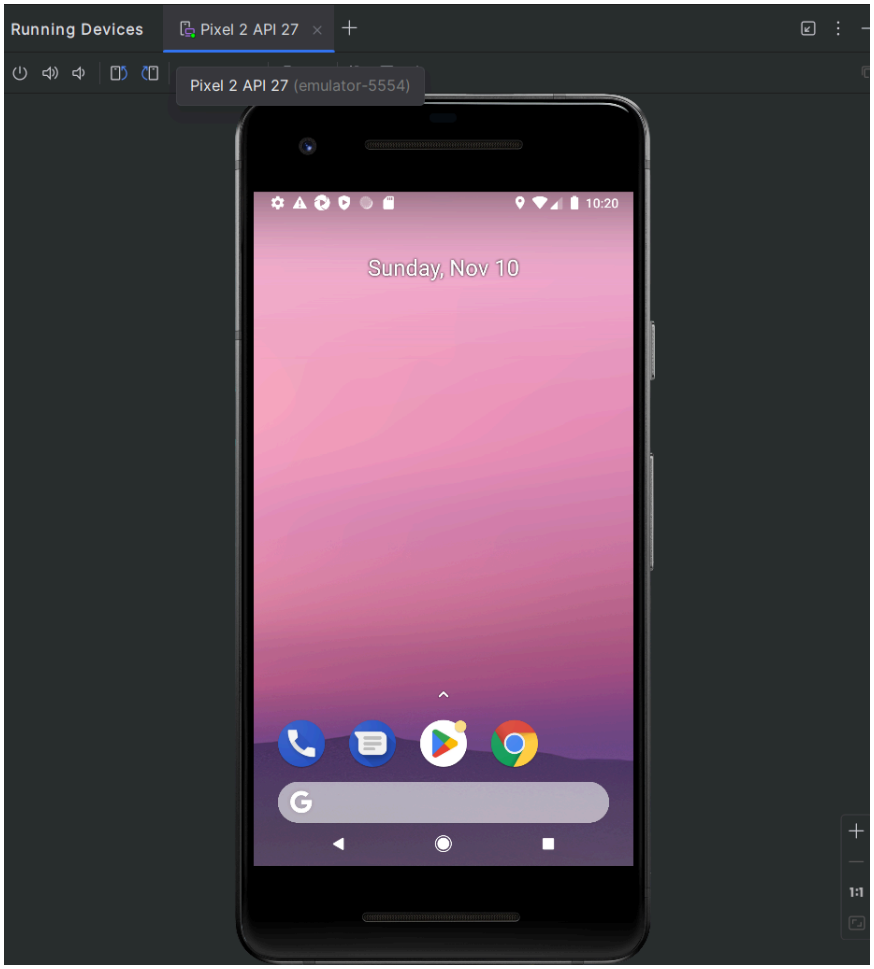
Steps:

1. **Wait for 2 seconds and click on the 'BACK HOME' button**: The test waits for 2 seconds and then clicks the 'BACK HOME' button.
2. **Click on the 'Menu' button**: Waits for the 'Menu' button to become visible and clicks it to navigate to the menu.
3. **Wait for 2 seconds and click on the 'LOGOUT' button**: The test waits for 2 seconds before clicking the 'LOGOUT' button to initiate the logout process.
4. **Wait for 5 seconds**: After clicking the 'LOGOUT' button, the test waits for 5 seconds to ensure the logout process completes.

5. Now Start the appium server in command prompt using “appium” command.

```
C:\Users\Aniket>appium
[Appium] Welcome to Appium v2.12.1
[Appium] The autodetected Appium home path: C:\Users\Aniket\.appium
[Appium] Attempting to load driver uiautomator2...
[Appium] Requiring driver at C:\Users\Aniket\.appium\node_modules\appium-uiautomator2-driver\build\index.js
[Appium] AndroidUiautomator2Driver has been successfully loaded in 1.361s
[Appium] Appium REST http interface listener started on http://0.0.0.0:4723
[Appium] You can provide the following URLs in your client code to connect to this server:
        http://192.168.109.42:4723/
        http://127.0.0.1:4723/ (only accessible from the same host)
[Appium] Available drivers:
[Appium]   - uiautomator2@3.8.1 (automationName 'UiAutomator2')
[Appium] No plugins have been installed. Use the "appium plugin" command to install the one(s) you want to use.
```

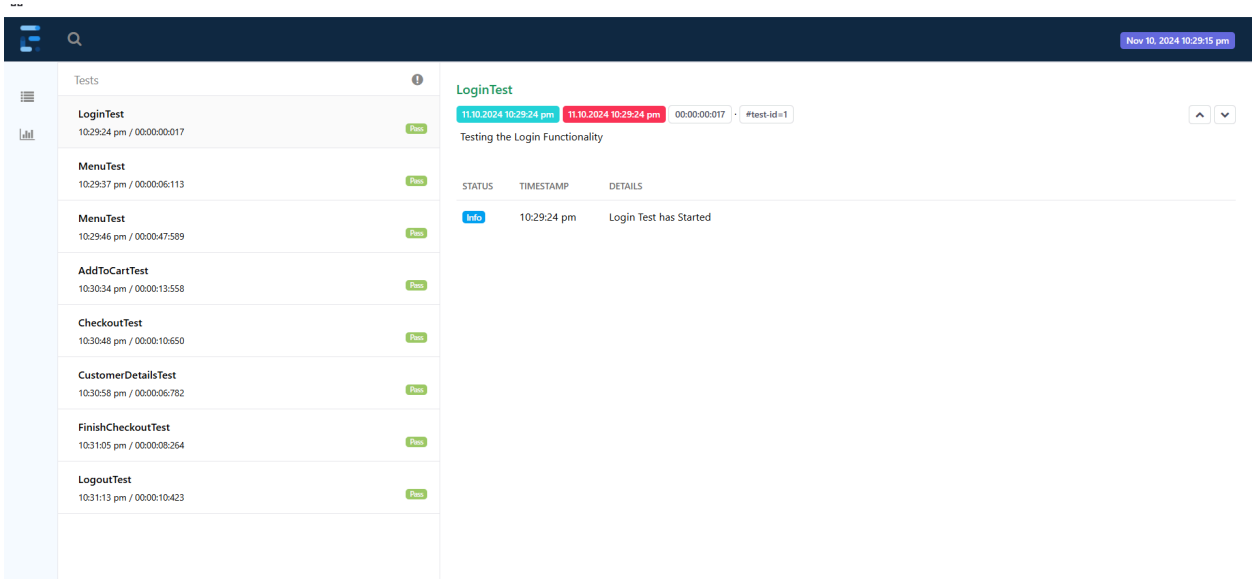
6. Start the emulator on android studio , here I am using Pixel_2_API_27.



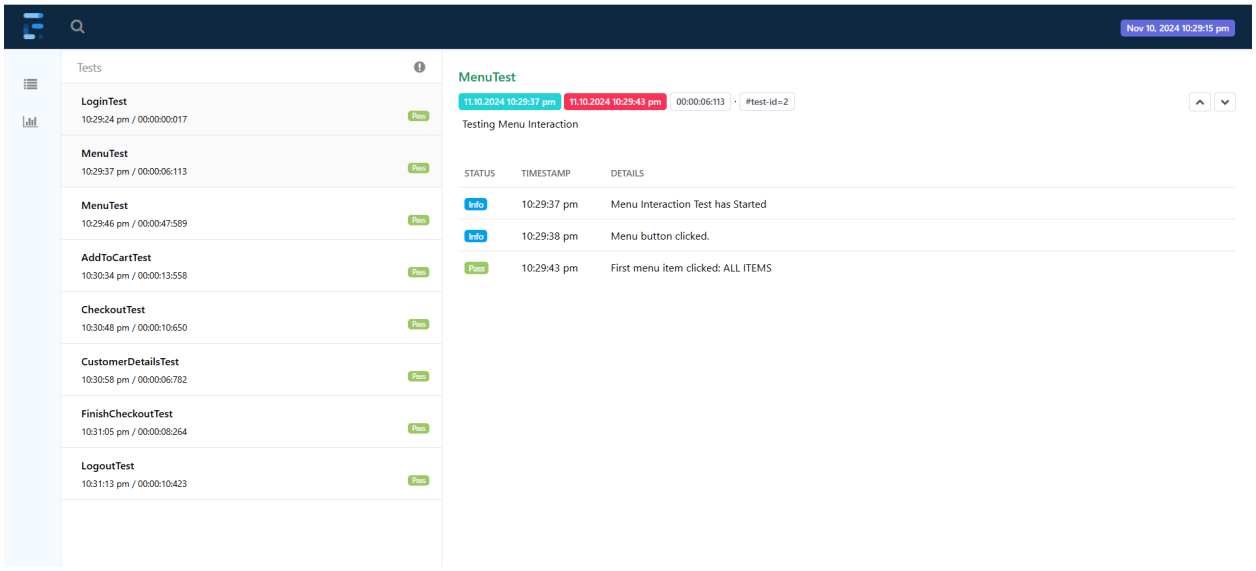
7. Now drag and drop the apk of the sauce labs sample e-commerce app.
8. After this, run the file (“TestCases”) in Eclipse IDE.

Output : Extent Report

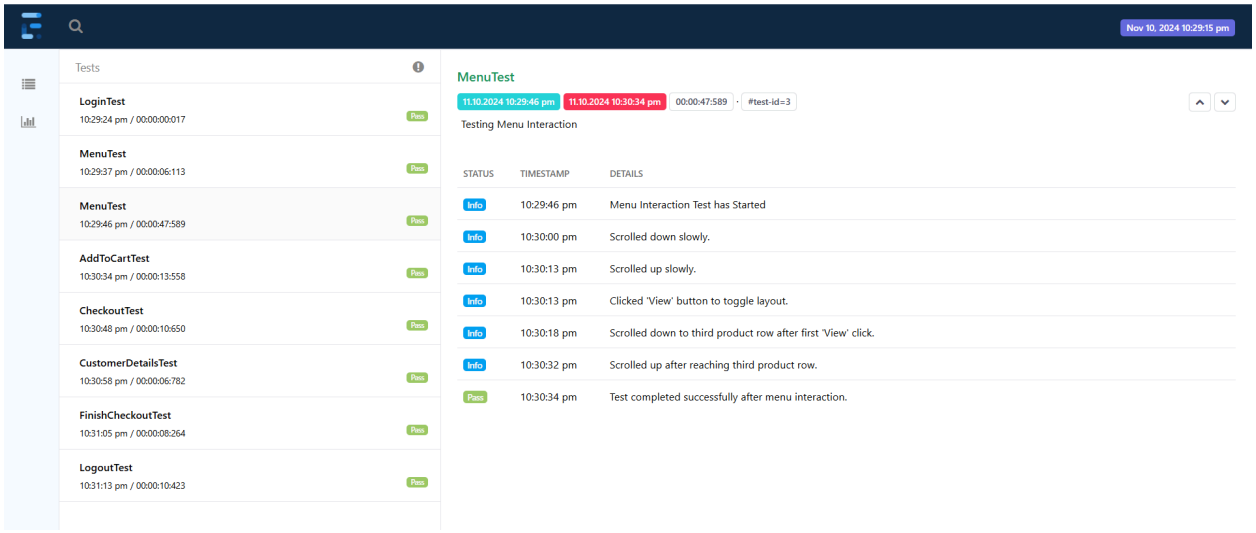
- **Test Case 1 :**



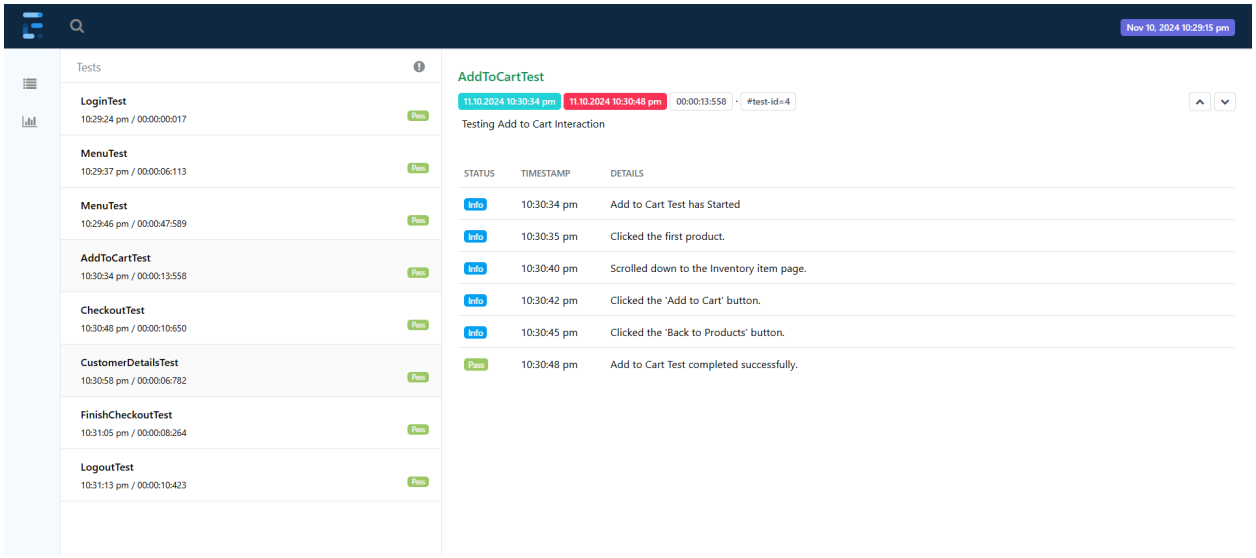
- **Test Case 2 :**



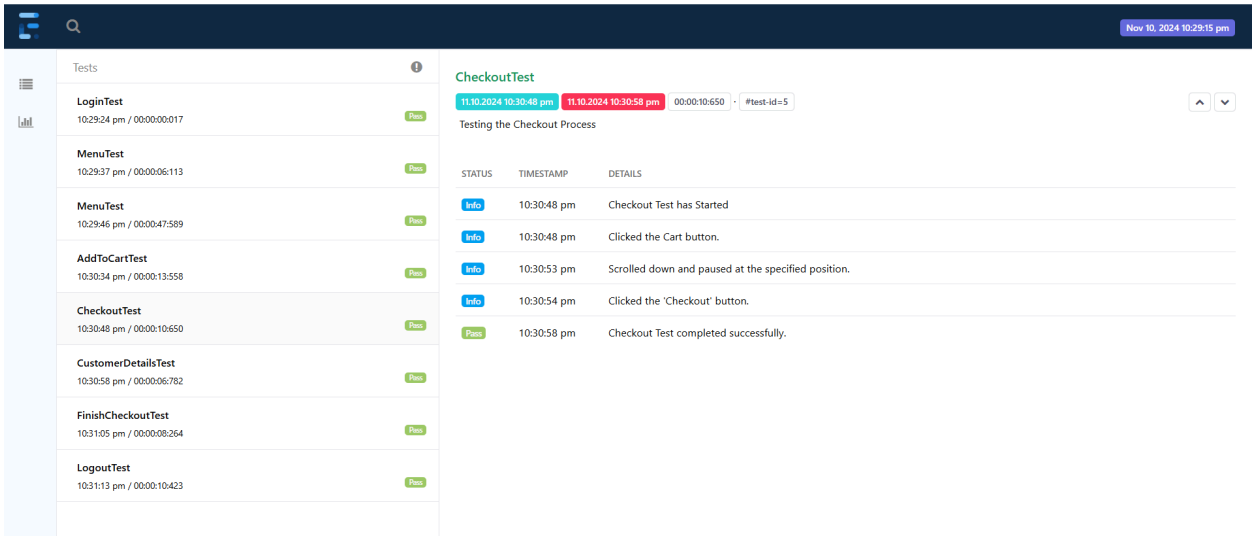
- **Test Case 3 :**



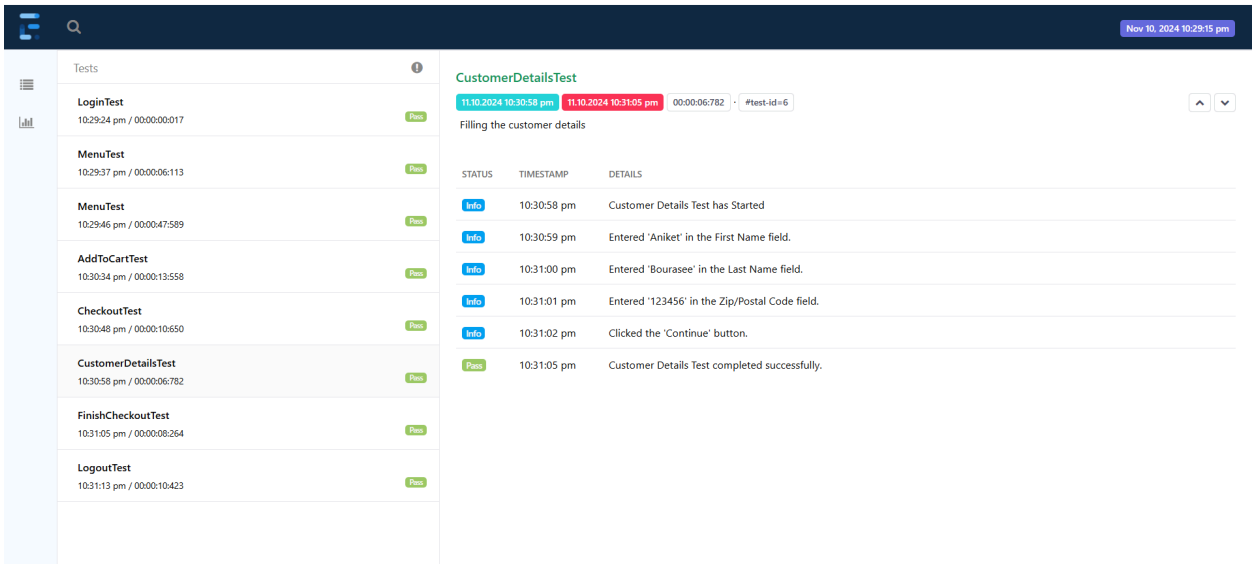
- **Test Case 4 :**



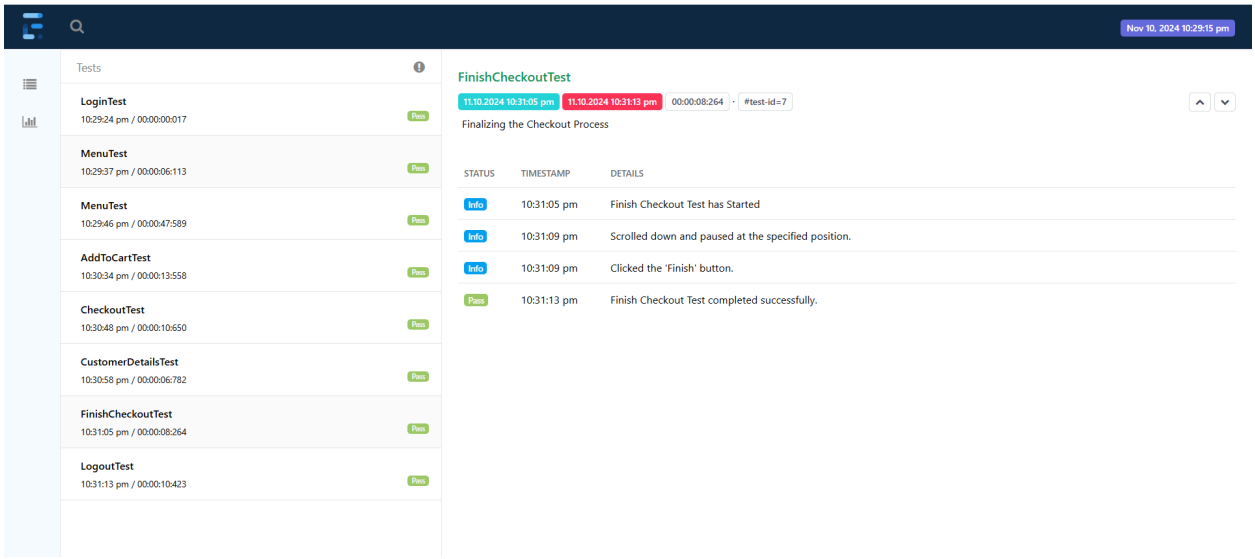
- **Test Case 5 :**



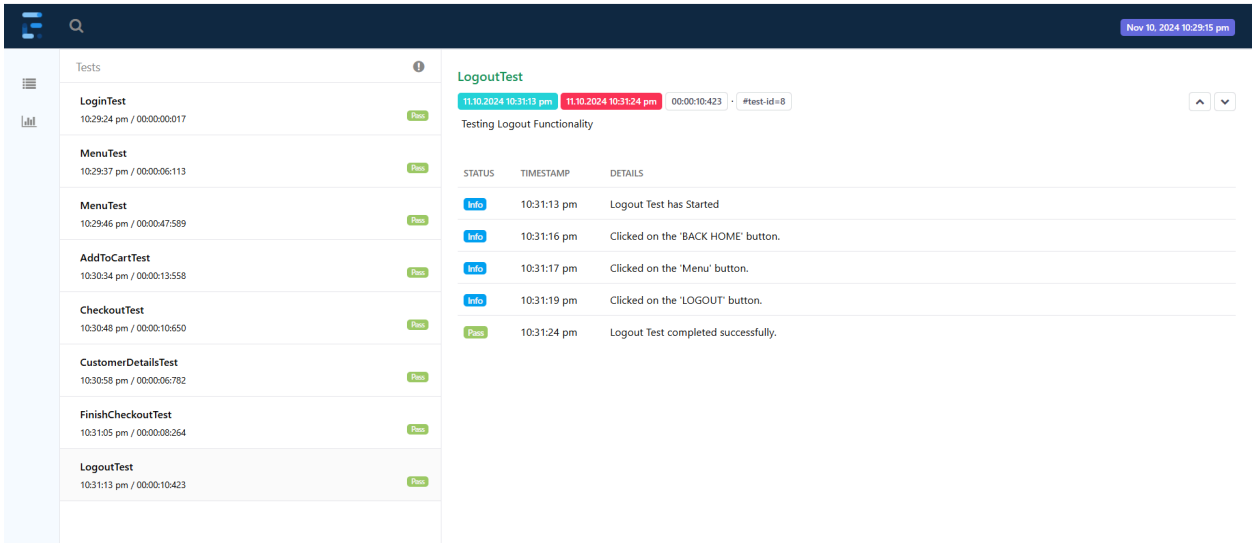
- **Test Case 6 :**



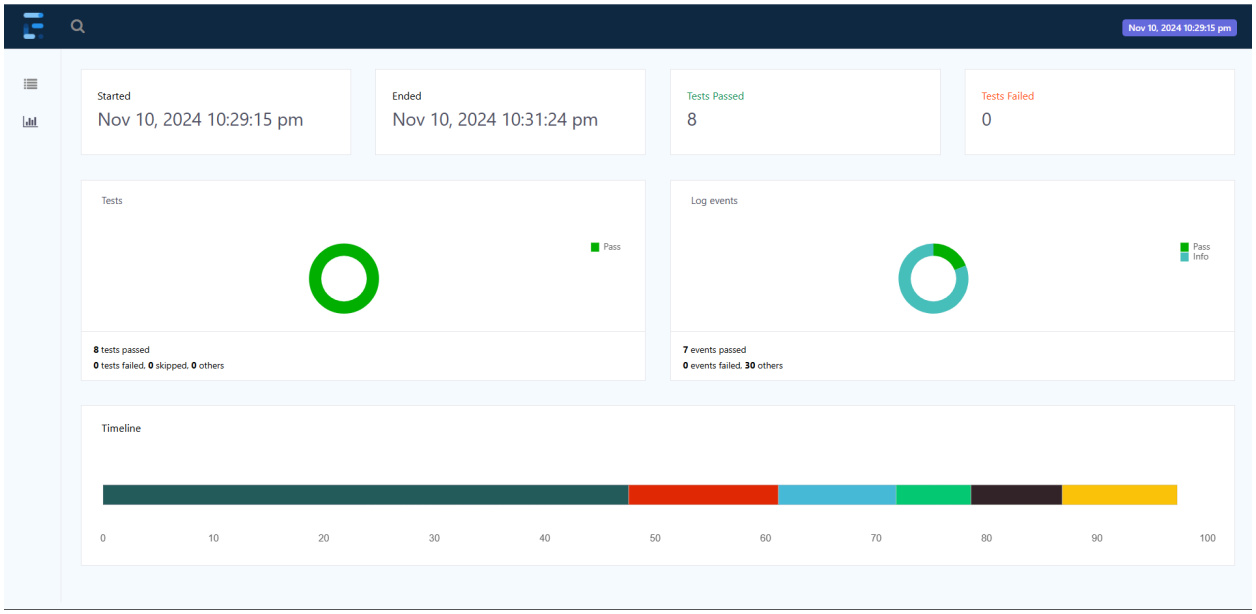
- **Test Case 7 :**



- **Test Case 8 :**



Overall Report :



Console Output :

```
[RemoteTestNG] detected TestNG version 7.4.0
Starting execution of testLogin.
Login screen appeared. Waiting for 3 seconds.
Clicked the 'standard_user' button.
Scrolled to the 'Username' field.
Waited for 2 seconds after scrolling.
Clicked the 'LOGIN' button.
Product screen appeared. Waiting for 3 seconds.
testLogin execution completed.
Starting execution of testMenuInteraction.
Waiting for Menu button to become visible.
Menu button is visible. Clicking on it.
Menu button clicked.
Waiting for 5 seconds before interacting with the menu.
Wait time completed.
Waiting for 'ALL ITEMS' menu option to become visible.
'ALL ITEMS' menu option is visible. Clicking on it.
'ALL ITEMS' menu option clicked.
Waiting for 3 seconds after navigating to 'ALL ITEMS' screen.
```

Wait time completed.
testMenuInteraction execution completed.
Starting execution of testProductInteraction.
Assuming menu interaction is already handled.
Starting to scroll down slowly.
Scrolling down step 1
Scrolling down step 2
Scrolling down step 3
Scrolling down step 4
Scrolling down step 5
Completed scrolling down slowly.
Starting to scroll up slowly.
Scrolling up step 1
Scrolling up step 2
Scrolling up step 3
Scrolling up step 4
Scrolling up step 5
Completed scrolling up slowly.
Waiting for 'View' button to become visible.
'View' button is visible. Clicking on it.
Clicked 'View' button.
Waiting for 2 seconds for UI layout to refresh.
Wait time completed.
Scrolling down to the third product row.
Scrolled down to the third product row.
Starting to scroll up after reaching the third product row.
Scrolling up step 1
Scrolling up step 2
Scrolling up step 3
Scrolling up step 4
Scrolling up step 5
Completed scrolling up after reaching the third product row.
Waiting for 2 seconds after scrolling up.
Wait time completed.
testProductInteraction execution completed.
Starting execution of testAddToCart.
Waiting for the first product to become visible.
First product is visible. Clicking on it.
Clicked the first product.
Waiting for the Inventory item page to become visible.

Inventory item page is visible. Scrolling down.
Scrolled down to the Inventory item page.
Waiting for 2 seconds before clicking the 'Add to Cart' button.
Wait time completed.
Waiting for the 'Add to Cart' button to become visible.
'Add to Cart' button is visible. Clicking on it.
Clicked the 'Add to Cart' button.
Waiting for 2 seconds after clicking the 'Add to Cart' button.
Wait time completed.
Waiting for the 'Back to Products' button to become visible.
'Back to Products' button is visible. Clicking on it.
Clicked the 'Back to Products' button.
Waiting for 3 seconds after clicking the 'Back to Products' button.
Wait time completed.
testAddToCart execution completed.
Starting execution of testCheckoutProcess.
Waiting for the Cart button to become visible.
Cart button is visible. Clicking on it.
Clicked the Cart button.
Waiting for 1 second before scrolling.
Scrolling down and pausing at the specified position.
Waiting for the 'Checkout' button to become visible.
'Checkout' button is visible. Clicking on it.
Clicked the 'Checkout' button.
Waiting for 4 seconds after clicking the 'Checkout' button.
Completed execution of testCheckoutProcess.
Starting execution of testFillCustomerDetails.
Waiting for the First Name field to become visible.
First Name field is visible. Entering 'Aniket'.
Entered 'Aniket' in the First Name field.
Waiting for the Last Name field to become visible.
Last Name field is visible. Entering 'Bourasee'.
Entered 'Bourasee' in the Last Name field.
Waiting for the Zip/Postal Code field to become visible.
Zip/Postal Code field is visible. Entering '123456'.
Entered '123456' in the Zip/Postal Code field.
Waiting for the 'Continue' button to become visible.
'Continue' button is visible. Clicking on it.
Clicked the 'Continue' button.
Waiting for 3 seconds after clicking the 'Continue' button.

```
Completed execution of testFillCustomerDetails.
Starting execution of testFinishCheckout.
Waiting for 1 second before scrolling.
Scrolling down and pausing at the checkout overview section.
Scrolled down and paused at the specified position.
Waiting for the 'Finish' button to become visible.
'Finish' button is visible. Clicking on it.
Clicked the 'Finish' button.
Waiting for 4 seconds after clicking the 'Finish' button.
Completed execution of testFinishCheckout.
Starting execution of testLogout.
Waiting for 2 seconds before clicking the 'BACK HOME' button.
Clicking the 'BACK HOME' button.
'BACK HOME' button clicked.
Waiting for the 'Menu' button to become visible.
'Menu' button is visible. Clicking on it.
'Menu' button clicked.
Waiting for 2 seconds before clicking the 'LOGOUT' button.
Clicking the 'LOGOUT' button.
'LOGOUT' button clicked.
Waiting for 5 seconds after clicking the 'LOGOUT' button.
Completed execution of testLogout.
PASSED: testLogin
PASSED: testFillCustomerDetails
PASSED: testFinishCheckout
PASSED: testMenuInteraction
PASSED: testCheckoutProcess
PASSED: testLogout
PASSED: testProductInteraction
PASSED: testAddToCart
===============================================
  Default test
  Tests run: 8, Failures: 0, Skips: 0
===============================================
===============================================
Default suite
Total tests run: 8, Passes: 8, Failures: 0, Skips: 0
===============================================
```