

Main Class:

```
package in.swiggy.pages;
import java.time.Duration;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
public class SwiggyMain {
    private WebDriver driver;
    public SwiggyMain() {
        driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.swiggy.com/");
    }
    public void enterDeliveryLocation(String location) {
        WebElement enterYourDeliveryLocation = driver.findElement(By.xpath("//input[@placeholder='Enter your delivery location']"));
        enterYourDeliveryLocation.sendKeys(location);
        try {
            Thread.sleep(2000);
            enterYourDeliveryLocation.sendKeys(Keys.ARROW_DOWN);
            enterYourDeliveryLocation.sendKeys(Keys.ENTER);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
    public void selectRestaurant() {
        WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
        WebElement firstRestaurant = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("(//img[contains(@alt,'Taj Mahal-Abids')])[1]")));
        firstRestaurant.click();
    }
    public void selectFirstDish() {
        WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
        WebElement firstDish = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("(//div[contains(@class, '_1RPOp')])[1]")));
        firstDish.click();
    }
    public void addToCart() {
        WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
        WebElement addToCartButton = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("(//div[contains(@class, '_1RPOp')])[1]")));
        // Use JavascriptExecutor to click the "Add to Cart" button directly
        JavascriptExecutor jsExecutor = (JavascriptExecutor) driver;
        jsExecutor.executeScript("arguments[0].click();", addToCartButton);
        try {
            Thread.sleep(2000); // Wait for the cart to update
        } catch (InterruptedException e) {
```

```

e.printStackTrace();
}
}

public void clickCheckout() {
WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
WebElement cartIcon = driver.findElement(By.xpath("//*[@id=\"root\"]/div[1]/header/div/div/ul/li[1]/div/div/a/span[2]"));
Actions actions = new Actions(driver);
actions.moveToElement(cartIcon).perform();
WebElement checkoutButton = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[contains(@class, '_55uP6')]")));
checkoutButton.click();
try {
Thread.sleep(2000); // Wait for the page to load
} catch (InterruptedException e) {
e.printStackTrace();
}
}

public void verifyLoginSignUpMessage() {
WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
WebElement verifyTextElement = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[contains(text(), 'To place your order now, log in to your existing account or sign up.')])));
if (verifyTextElement.isDisplayed()) {
System.out.println("Verification successful! Text is visible on the next page.");
} else {
System.out.println("Verification failed! Text is not visible on the next page.");
}
}

public void closeBrowser() {
driver.quit();
}

public static void main(String[] args) throws InterruptedException {
SwiggyMain swiggyMain = new SwiggyMain();
swiggyMain.enterDeliveryLocation("Hyderabad");
swiggyMain.selectRestaurant();
swiggyMain.selectFirstDish();
swiggyMain.addToCart();
swiggyMain.clickCheckout();
swiggyMain.verifyLoginSignUpMessage();
swiggyMain.closeBrowser();
}
}

```

Before And After Hooks:

```

package in.swiggy.teststeps;
import io.cucumber.java.After;
import io.cucumber.java.Before;
import org.openqa.selenium.chrome.ChromeDriver;
public class BeforeAfter {
protected static ChromeDriver driver;
@Before
public void setUp() {

```

```

// Initialize WebDriver before each scenario
driver = new ChromeDriver();
driver.manage().window().maximize();
}
}

@After
public void tearDown() {
// Quit WebDriver after each scenario
if (driver != null) {
driver.quit();
}
}
}

```

Tools:

```

package in.swiggy.teststeps;
import org.openqa.selenium.WebDriver;
public class Tools {
protected static WebDriver driver;
}

```

Test Runner Class:

```

package in.swiggy.teststeps;
import io.cucumber.testng.AbstractTestNGCucumberTests;
import io.cucumber.testng.CucumberOptions;
@CucumberOptions(
features = "Features",
glue = "in.swiggy.teststeps"
//tags = "@TC_101"
)
public class TestRunner extends AbstractTestNGCucumberTests{
}

```

Driver Class:

```

package in.swiggy.teststeps;
import org.openqa.selenium.chrome.ChromeDriver;
import in.swiggy.pages.SwiggyMain;
public class Driver {
protected static ChromeDriver driver;
protected static SwiggyMain swiggyMain;
public static void init() {
driver = new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.swiggy.com");
swiggyMain = new SwiggyMain();
}
public static void closeDriver() {
driver.quit();
}
}

```

Test Class:

```
package in.swiggy.teststeps;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.When;
import in.swiggy.pages.SwiggyMain;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Then;
public class SwiggyTestSteps {
private SwiggyMain swiggyMain;
@Given("I am on the Swiggy website")
public void openSwiggyWebsite() {
swiggyMain = new SwiggyMain();
}
@When("I enter my delivery location as {string}")
public void enterDeliveryLocation(String location) {
swiggyMain.enterDeliveryLocation(location);
}
@And("I select a restaurant")
public void selectRestaurant() {
swiggyMain.selectRestaurant();
}
@And("I select the first dish available")
public void selectFirstDish() {
swiggyMain.selectFirstDish();
}
@And("I add the dish to the cart")
public void addToCart() {
swiggyMain.addToCart();
}
@And("I click on the cart icon to check out")
public void clickCheckout() {
swiggyMain.clickCheckout();
}
@Then("I should see the message to log in or sign up")
public void verifyLoginSignUpMessage() {
swiggyMain.verifyLoginSignUpMessage();
}
}
```

Feature File:

```
Feature: Swiggy Order Placement
@TC-101
Scenario: Place an order on Swiggy
Given I am on the Swiggy website
When I enter my delivery location as "Hyderabad"
And I select a restaurant
And I select the first dish available
And I add the dish to the cart
And I click on the cart icon to check out
Then I should see the message to log in or sign up
```

Testng.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
<listeners>[REDACTED]
<listener class-name="com.Listeners.TestListener"></listener>
</listeners>[REDACTED]
<test thread-count="5" name="Test">
<classes>[REDACTED]
<class name="in.swiggy.teststeps.TestRunner"></class>
</classes>[REDACTED]
</test> <!-- Test -->
</suite> <!-- Suite -->
```

Testng Report:

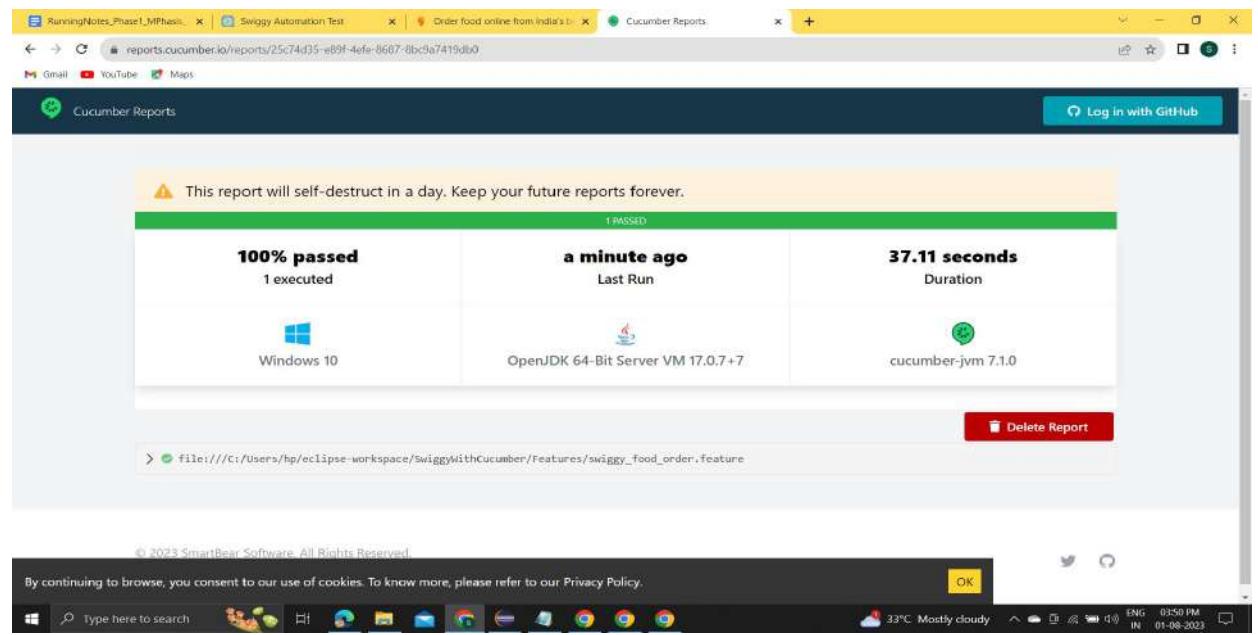
The screenshot shows a web browser window displaying a test automation report. The title bar reads "Test-Automation-Report.html". The main content area has a header "Extent" and a sub-header "Test-Automation-Report.html". Below this, there are two main sections: "Tests" and "runScenario".

Tests:
1 test(s) passed
0 test(s) failed, 0 others

runScenario:
Tuesday, August 01, 2023, 03:39 PM (IST) | Tuesday, August 01, 2023, 03:39 PM (IST) | 0h 0m 33s+0ms

Status	Timestamp	Details
Pass	3:39:33 PM	Test passed

Cucumber Report:



The screenshot shows a Cucumber report page with the following details:

- Warning:** This report will self-destruct in a day. Keep your future reports forever.
- 1 PASSED**
- 100% passed**
1 executed
- a minute ago**
Last Run
- 37.11 seconds**
Duration
- Windows 10**
- OpenJDK 64-Bit Server VM 17.0.7+7**
- cucumber-jvm 7.1.0**
- Delete Report** button

At the bottom, there is a cookie consent message: "By continuing to browse, you consent to our use of cookies. To know more, please refer to our Privacy Policy." with an "OK" button. The taskbar at the bottom shows various application icons and the system clock.