



# VIT<sup>®</sup>

**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

## SCHOOL OF COMPUTER SCIENCE ENGINEERING AND INFORMATION SYSTEMS

Winter Semester 2023-24

### **TITLE : SHOPPERSSTOP APPLICATION**

Programme/Branch: M. Tech(SE)

Course Name: Software Testing

Course Code: SWE2005

Faculty: **DR. ARIVUSELVAN K**

Submitted by:

S.NO	NAME	REG NO
1	Siddharth Jain	22MIS0197
2	M Prasanth	22MIS0201
3	Sanjay S	22MIS0419

## **1) ABSTRACT:**

Testing the Shoppers Stop website in software testing is crucial to ensure that it meets the quality standards expected by the users. The purpose of testing is to identify and fix any defects or issues that may affect the website's performance, usability, security, compatibility, and accessibility. By testing the website, software testers can ensure that it functions correctly, provides a seamless shopping experience, and protects the user's personal and financial information. Additionally, testing helps to enhance the website's credibility, reputation, and profitability by reducing the risk of errors, customer complaints, and business losses. Therefore, software testing plays a significant role in ensuring the success and sustainability of the Shoppers Stop website.

## **2) LITERATURE SURVEY:**

### **1. Study of the Testing of E-Commerce Applications:**

**Samkit Jain and Pradnya Purandare 2021 J. Phys.: Conf. Ser. 1964:**

Usability Testing determines the friendliness of the application to meet customers' requirements. It must be adopted in the early stage of SDLC (Software Development Life Cycle) to avoid issues in a later stage. Today, E-commerce has revolutionized the mode of shopping, and thus, we need to focus on ways to improve the usability of e-commerce applications. We opted for survey questionnaires as our research methodology, along with testing websites using online tools based on the four parameters Connectivity,

Readability, Accessibility, and Functional Performance. In the survey, the four most popular e-commerce applications were Amazon, Flipkart, Big basket, and Paytm. Out of which, Amazon was the most used e-commerce application by the consumers. Although, testing tools are depicted that applications need to improve in certain areas. From the analysis, we concluded that consumers are aware of the upcoming Jio-Mart, which can give tough competition to existing companies. Thus, all the existing e-commerce companies need to improve their web and mobile applications to retain their customer base.

## **2.E-COMMERCE SITE FOR ONLINE object sales.**

Electronic Commerce (e-commerce) applications support the interaction between different parties participating in a commerce transaction via the network, as well as the management of the data involved in the process .

The increasing importance of e-commerce is apparent in the study conducted by researchers at the GVU (Graphics, Visualization, and Usability) Centre at the Georgia Institute of Technology. In their summary of the findings from the eighth survey, the researchers report that “e-commerce is taking off both in terms of the number of users shopping as well as the total amount people are spending via Internet based transactions”.

### **3.ONLINE SHOPPING testing**

#### **B.E., Osmania University, 2005:**

The main objective of this application is to make it interactive and its ease of use. It would make searching, viewing and selection of a product easier. It contains a sophisticated search engine for users to search for products specific to their needs. The search engine provides an easy and convenient way to search for products where a user can Search for a product interactively and the search engine would refine the products available based on the user's input. The user can then view the complete specification of each product. They can also view the product reviews and also write their own reviews. The application also provides a drag and drop feature so that a user can add a product to the shopping cart by dragging the item in to the shopping cart. The main emphasis lies in providing a user-friendly search engine for effectively showing the desired results and its drag and drop behavior.

### **4. e-commerce websites performance and maintainance and demands survey.**

With the rapid development of technology and the economy, the expansion of the network has had a huge impact on the rapid expansion of the industrial agglomeration e-commerce industry, as well as ensuring the shopping experience of consumers. The rapid expansion of industrial cluster e-commerce has precisely avoided the limitations of logistical bottlenecks. Current networks and modern information technologies can provide good support and maintain a huge growth potential. In addition, digital

technologies such as multimedia are becoming increasingly important in industry cluster marketing, and the concept of industry cluster e-commerce models is gaining more and more attention from companies. However, virtual e-commerce systems under industrial clusters have not been well researched in the existing studies. In this paper, through extensive research, literature reading and website browsing statistics, the virtual e-commerce models of different industrial agglomerations are studied. Firstly, the concept of big data and the processing of big data are given. Secondly, the concept of industrial agglomeration and the relationship between industrial agglomeration and e-commerce are analysed.

### **3) APPLICATION OVERVIEW:**

The Shoppers Stop website in software testing is to ensure that the website meets the business requirements and user expectations. The website should be tested thoroughly to ensure that it provides a seamless and enjoyable shopping experience to its users.

Here are some specific areas that need to be tested as part of the application overview of testing the Shoppers Stop website:

#### **Functionality:**

The website's functionality should be tested to ensure that all the features are working correctly, such as browsing products, adding them to the cart, placing orders, and making payments. Any defects or issues should be identified and resolved.

**Usability:**

The website's usability should be tested to ensure that it is user-friendly, easy to navigate, and the design is aesthetically pleasing. The user experience should be smooth and intuitive.

**Performance:**

The website's performance should be tested to ensure that it is fast and responsive, even during high traffic periods. The website should load quickly and should not experience any downtime or crashes.

**Security:**

The website's security should be tested to ensure that it is secure and protects users' personal and financial information. The website should be protected against hacking attempts and other security threats.

**Compatibility:**

The website's compatibility should be tested to ensure that it works on different devices, browsers, and operating systems. The website should be tested on various platforms and devices to ensure that it works as expected.

Overall, the application overview of testing the Shoppers Stop website in software testing is to ensure that the website is reliable, secure, and user-friendly. The website should provide an excellent shopping experience to its users, which is essential for the success of the business.

## **FEATURES OF APPLICATION:**

Shoppersstop is an e-commerce application that provides an online shopping platform for customers. The application has a wide range of features that are designed to enhance the user experience and make shopping more convenient.

Here are some of the key features of Shoppersstop:

### **Product Catalog:**

Shoppersstop provides a comprehensive catalog of products, including clothing, accessories, beauty products, home decor, and more. Users can easily browse through the categories, filter by size, color, price, brand, and other attributes, and view detailed product information, images, and customer reviews.

### **Search and Navigation:**

The application provides a powerful search engine that allows users to find products quickly and easily. Users can search by keywords, product name, brand, category, or other criteria, and the application will return relevant results. Navigation is also made easy with a user-friendly interface, including breadcrumb trails and category listings.

### **Cart and Checkout:**

Shoppersstop has a simple and intuitive cart and checkout process. Users can add items to their cart, view the total cost of their order, apply promo codes or coupons, and choose from a range of payment and shipping options.

### **Ratings and Reviews:**

Users can rate products and leave reviews, which are displayed on the product pages for other users to see. This feature helps build trust and credibility, and also provides valuable feedback to the sellers.

### **Customer Support:**

Shoppersstop provides excellent customer support, including a live chat feature, email support, and a toll-free helpline. Users can get help with their orders, returns, exchanges, or any other issues they may encounter.

### **Social Media Integration:**

The application has seamless integration with social media platforms like Facebook, Twitter, and Instagram. Users can share products with their friends and followers or follow the brand's social media pages for updates and promotions.

## **4) PROJECT SCOPE:**

The Shopper-Stop E-commerce website, which is an online platform for users to purchase products and services. The testing will be carried out to ensure the website is functional, user-friendly, and meets the requirements and expectations of the end-users.

### **In Scope:**

#### **Functionality testing:**

The website's features and functionality will be thoroughly tested to ensure they work as expected, including



product search, browsing, adding to cart, checkout, payment processing, and order tracking.

### **Usability testing:**

The website's user interface, design, and layout will be tested to ensure that it is user-friendly, easy to navigate, and provides a seamless experience to users.

### **Performance testing:**

The website's performance will be tested to ensure that it can handle many concurrent users and transactions without crashing or slowing down.

### **Security testing:**

The website's security measures will be tested to ensure that user data is protected and secure from unauthorized access, hacking, or data breaches.

### **Compatibility testing:**

The website will be tested for compatibility with various browsers, devices, and operating systems to ensure that it works consistently across different platforms.

### **Out of Scope:**

#### **Localization testing:**

Testing the website's functionality in different languages or regions is not within the scope of this testing project.

#### **Third-party integrations:**

Testing third-party integrations with the website, such as payment gateways or social media platforms, is not within the scope of this testing project.

**Backend testing:**

Testing the website's backend infrastructure, including databases, servers, and APIs, is not within the scope of this testing project.

Items not tested in Shopper-Stop Website:

**Third-party applications:**

Any third-party applications that are integrated with the website will not be tested in this project.

**Mobile applications:**

Mobile applications for the website will not be tested in this project.

**Physical products:**

The physical products sold through the website will not be tested in this project.

This testing scope document outlines the various areas that will be tested for the Shopper-Stop E-commerce website, as well as the areas that are out of scope. It is important to note that this document is subject to change based on the project's requirements and the client's expectations.

**5) METRICS:**

Test metrics are indicators of the efficiency, effectiveness, quality, and performance of software testing techniques. These metrics allow professionals to collect data about various testing procedures and devise ways to make

them more efficient. It also allows them to improve testing techniques in various aspects of a product, such as its quality, security, customer satisfaction, development cost and performance. The life cycle of creating a software testing metric is:

### **Analysis:**

Testing professionals devise a metric to test the effectiveness of their testing methods.

### **Communication:**

This stage involves presenting the metric to the management and stakeholders for approval.

### **Evaluation:**

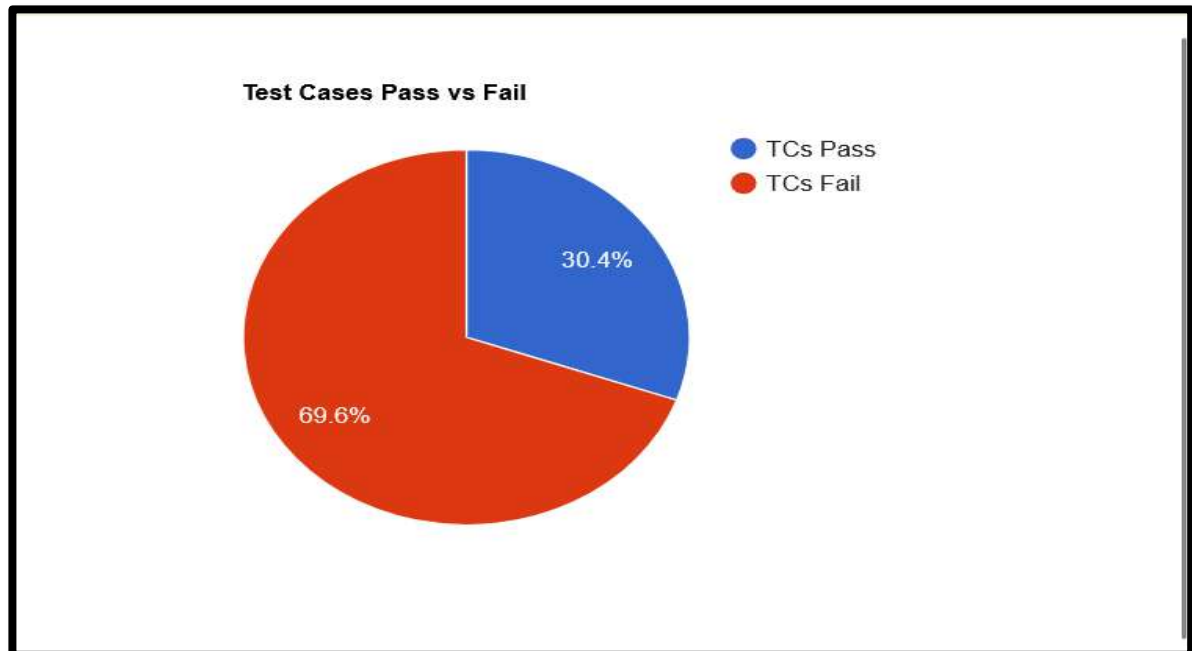
Professionals evaluate the metric using an extensive data set in this step.

### **Report:**

This step involves creating a detailed report on the metric, including its formula, experiments, and results.

Test cases planned	Test cases executed	Test cases passed	Test cases failed
35	23	7	16

Test case Type	Value
TCs Pass	7
TCs Fail	16



## 6) TYPES OF TESTING PERFORMED:

### Robustness Testing:

Robustness testing is a type of testing that is performed to assess the ability of a system or component to function correctly when it is subjected to invalid or unexpected inputs, or when it is operating outside of its specified operating conditions.

The purpose of robustness testing is to identify the parts of the system that are most vulnerable to failure and to determine how the system can be made more resistant to failure.

The purpose of robustness testing is to identify the parts of the system that are most vulnerable to failure and to

determine how the system can be made more resistant to failure.

The purpose of robustness testing is to identify the parts of the system that are most vulnerable to failure and to determine how the system can be made more resistant to failure.

### **Decision Table Testing:**

Decision table testing is a software testing technique used to test system behaviour for different input combinations. This is a systematic approach where the different input combinations and their corresponding system behaviour (Output) are captured in a tabular form. That is why it is also called as a Cause-Effect table where Cause and effects are captured for better test coverage.

A Decision Table is a tabular representation of inputs versus rules/cases/test conditions.

It is a very effective tool used for both complex software testing and requirements management.

A decision table helps to check all possible combinations of conditions for testing and testers can also identify missed conditions easily. The conditions are indicated as True(T) and False(F) values.

### **Unit Testing:**

Unit Testing is a software testing technique by means of which individual units of software i.e. group of computer program modules, usage procedures, and operating procedures are tested to determine whether they are suitable for use or not. It is a testing method using which every independent

module is tested to determine if there is an issue by the developer himself.

Unit testing is defined as a type of software testing where individual components of a software are tested.

The objective of Unit Testing is:

- To isolate a section of code.

- To verify the correctness of the code.

- To test every function and procedure.

- To fix bugs early in the development cycle and to save costs.

- To help the developers to understand the code base and enable them to make changes quickly.

- To help with code reuse.

### **Equivalence Testing and Boundary value testing:**

Boundary testing is the process of testing between extreme ends or boundaries between partitions of the input values.

So, these extreme ends like Start- End, Lower- Upper, Maximum-Minimum, Just Inside-Just Outside values are called boundary values and the testing is called “boundary testing”.

Equivalence Partitioning or Equivalence Class Partitioning is type of black box testing technique which can be applied to all levels of software testing like unit, integration, system, etc. In this technique, input data units are divided into equivalent partitions that can be used to derive

test cases which reduces time required for testing because of small number of test cases.

It divides the input data of software into different equivalence data classes.

You can apply this technique, where there is a range in the input field.

### **Performance Testing:**

Performance Testing is a software testing process used for testing the speed, response time, stability, reliability, scalability, and resource usage of a software application under a particular workload. The goal of Performance Testing is not to find bugs but to eliminate performance bottlenecks. Performance Testing is done to provide stakeholders with information about their application regarding speed, stability, and scalability.

Determines whether the application responds quickly

Determines the maximum user load the software application can handle.

Determines if the application is stable under varying loads.

## **7) PROBLEM STATEMENT:**

The Shoppers Stop website in software testing is to ensure that the website functions as intended and provides an excellent user experience. The website should be tested for its functionality, usability, performance, security, compatibility, and accessibility.

The problem is to identify any issues, defects, or bugs in the website that may affect its functionality, usability, or security. These issues could include problems with navigation, broken links, slow page load times, security vulnerabilities, or compatibility issues with different devices and browsers.

The problem is also to ensure that the website meets the expectations and requirements of its users. Users should be able to easily find and purchase the products they are looking for, make secure payments, and receive excellent customer service.

The problem is to ensure that the website is ready for release to the public and that it meets the quality standards of the company. By testing the website thoroughly, the company can identify and address any issues before they impact the user experience and the reputation of the brand.

Overall, the problem statement for testing the Shoppers Stop website is to ensure that it is functional, user-friendly, secure, and compatible, and that it meets the needs and expectations of its users.

## **8) SOFTWARE REQUIREMENTS SPECIFICATION:**

### **1.Introduction**

#### **1.1. Brief overview of Work**

Testing the Shoppers Stop website involves checking its functionality, usability, performance, security, compatibility, and accessibility to ensure that it works correctly, is user-friendly, and provides an excellent shopping experience. Software testers identify and fix any defects or issues before the website is released to the public. This ensures that users



have a smooth and seamless shopping experience, which is essential for the success of the business. The testing process involves checking that all features are working correctly, the website is easy to navigate, fast and responsive, secure, works on different devices and browsers, and accessible to all users.

## **1.2. Objective**

The objective of testing the Shoppers Stop website is to ensure that it meets the expectations and requirements of its users. The testing process aims to identify and eliminate any defects, bugs, or issues that could affect the website's functionality, performance, security, usability, compatibility, or accessibility.

By conducting thorough testing, the website's stakeholders can ensure that it provides a seamless and enjoyable shopping experience to its users. This, in turn, can improve customer satisfaction and loyalty, boost sales, and enhance the overall reputation and success of the business. Additionally, testing helps to reduce the risk of costly errors and negative impact on the business's revenue and brand image.

## **1.3. Scope**

The scope of testing the Shoppers Stop website involves verifying and validating various aspects of the website to ensure its quality, reliability, and performance. This includes testing the functionality of the website, such as the ability to browse products, add items to the cart, and checkout. It also involves testing the usability of the website, such as the ease of navigation, responsiveness, and user interface design.

Additionally, the website must be tested for its security, compatibility, and accessibility. The testing process includes manual and automated testing to detect any defects or bugs in the website and to ensure that it meets the business and user requirements. The goal of testing the Shoppers Stop website is to deliver a high-quality website that meets the expectations of the users and enhances their shopping experience.

## **2. PRODUCT FUNCTIONS**

Load testing is a type of performance testing that simulates a real-world load on any software, application, or website.

That's why we build tools like retrace to help you monitor application performance and fix bugs before your code ever gets to production.

Load testing examines how the system behaves during normal and high loads and determines if a system, piece of software, or computing device can handle high loads given a high demand of end-users.

This tool is typically applied when a software development project nears completion.

The main objective of using the load testing-

The load testing is used to perform the maximum quantity of software applications without important performance breakdown.

It is used to identify the total count of users that can access the application simultaneously.

### **3. SPECIFICATION REQUIREMENTS**

#### **3.1. Software requirements**

Database Server: Microsoft SQL Server

Web Server: Internet Information Server, Apache HTTP server

Technologies: MySQL, PHP, SQL, XAMPP

#### **3.2. Functional requirements**

##### **User Registration:**

Users should be able to register for a new account on the website by providing their personal details, such as name, email address, phone number, and password. The website should validate the information and display an error message if any details are missing or invalid.

##### **Product Search:**

Users should be able to search for products on the website using keywords, categories, or filters. The website should display relevant search results and allow users to sort them by price, popularity, or other criteria.

##### **Product Details:**

Users should be able to view detailed information about each product, including its name, description, price, images, reviews, and ratings.

##### **Checkout:**

Users should be able to checkout their items by providing their shipping and billing information, selecting a payment method, and confirming their order. The website

should display a summary of the order and allow users to review and edit their information before finalizing the purchase.

### **Payment Processing:**

The website should process payments securely and support multiple payment methods, such as credit cards, debit cards, net banking, and digital wallets. The website should display a confirmation message after the payment is processed successfully.

### **Order Tracking:**

Users should be able to track their order status on the website by providing their order number or email address. The website should display the status of the order, estimated delivery date, and shipping details.

### **Customer Support:**

The website should provide customer support through multiple channels, such as email, phone, or live chat. The website should also have a FAQ section to help users find answers to common questions.

### **User Profile:**

Users should be able to view and edit their profile information, such as name, email address, phone number, and password. The website should also allow users to view their order history and track their loyalty points or rewards.

### **3.3. Non-Functional Requirements**

#### **Performance:**

The application should be fast and responsive, with pages loading quickly and without delay. Response times should be consistent, even during peak usage periods.

#### **Scalability:**

The application should be able to handle many users and transactions without slowing down or crashing. It should be designed to scale up or down as usage patterns change.

#### **Availability:**

The application should be available 24/7, with minimal downtime for maintenance or upgrades. Any scheduled downtime should be communicated to users in advance.

#### **Reliability:**

The application should be reliable, with minimal errors or crashes. Any errors or issues should be logged and quickly addressed by the development team.

#### **Security:**

The application should be secure, with strong measures in place to protect user data and prevent unauthorized access. This includes secure storage of user data, encryption of sensitive data, and secure payment processing.

#### **Usability:**

The application should be easy to use and navigate, with an intuitive user interface and clear instructions for users.

It should be accessible to users with different levels of technical expertise.

### **Compatibility:**

The application should be compatible with different devices, operating systems, and browsers, to ensure a consistent experience for users. This includes testing on different platforms and configurations.

### **Maintainability:**

The application should be easy to maintain, with clear documentation and code that is easy to modify and update. It should also be designed to minimize the risk of introducing new errors or issues during updates or modifications.

### **Performance under load:**

The application should be able to perform well under heavy load, with consistent response times and no errors or crashes. This includes testing under realistic conditions to simulate high usage periods.

## **4. BLACK BOX TESTING**

Black box testing is a software testing technique that focuses on testing the external behaviour of a software system without any knowledge of its internal workings. In other words, the tester has no information about how the system is designed or implemented, and tests the system only based on its functional specifications.

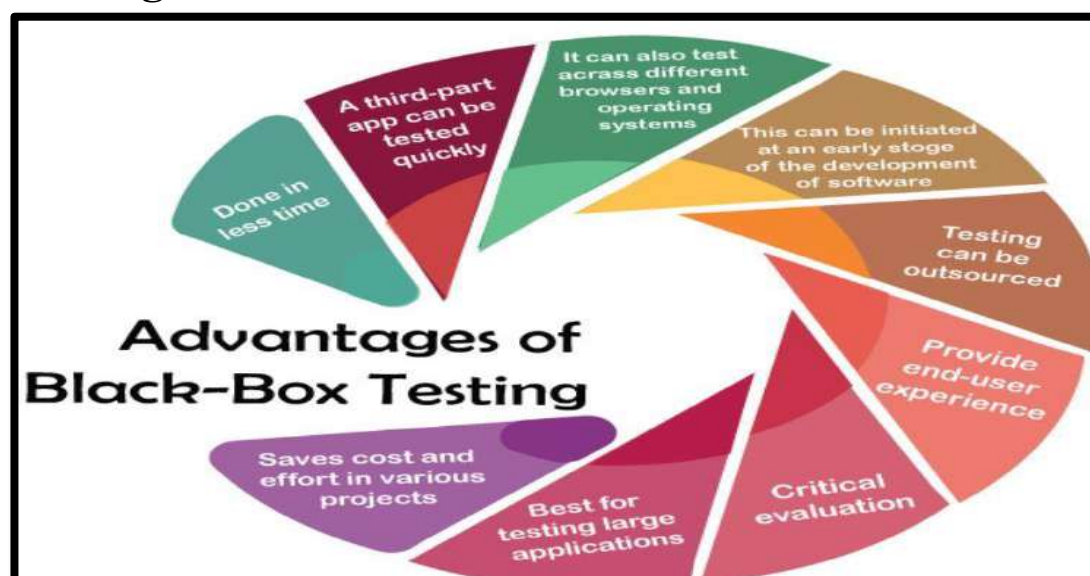
During black box testing, testers create test cases based on the system's requirements or specifications. They execute these test cases on the system to verify if it behaves as expected and meets the functional requirements. The goal is to uncover defects and defects that could affect the user's experience or cause the system to fail.

Black box testing can be performed at different levels of testing, including unit testing, integration testing, system testing, and acceptance testing. It can be conducted manually or using automated tools.

### **Types of Black Box testing:**

1. Boundary Value Analysis (BVC).
2. Robustness testing.
3. Equivalence Partitioning.
4. Decision Table testing.
5. State Based testing.
6. Cause Effect Based testing.

### **Advantages:**



## 4.1. Decision table-based Testing:

### Condition Stubs:

C1: User\_id

C2: Password

C3: Blank User\_id/Password

### Action Stubs:

A1: Success

A2: Error Message

A3: Prompt a message to fill the User\_id/Password blank

	STATE	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
Condition Sub	User_id	-	N	N	Y	Y
	Password	-	N	Y	N	Y
	Blank User_id/Password	Y	N	N	N	N
Action Sub	Success					✓
	Error Message		✓	✓	✓	
	Prompt a message to fill the User_ID/Password blank	✓				



Here,

**N** : Invalid input

**Y** : Valid input

✓ : Identifies the output

### Designing Test Cases:

Test case Id	Description	Input		Expected Output	Rule Covered
		User_ID	Password		
TC01	Not entered password	Sanjay26	-	Prompt message to fill the blank password	1
TC02	Entered wrong user_id and password	ABC	123	Error message	2
TC03	Entered wrong password	Prasanth1	123	Error message	4
TC04	Entered wrong user_id	ABC	Abc@123	Error message	3
TC05	Entered correct credentials	Ganesh12	Abc@78	Success	5
TC06	Not entered user id	-	Gfd@45	Prompt message	1

				to fill user id	
TC07	Entered wrong user_id and password	ABCD	A12345	Error message	2
TC08	Entered wrong password	Sid12345	A12	Error message	4
TC09	Entered wrong user_id	Abcde	Abc@200	Error message	3
TC10	Entered correct credentials	Sidd1278	Anx@234	Success	5

### Text Execution:

Test cases planned	Test cases executed	Test cases passed	Test cases failed
15	10	2	8

## 4.2 Boundary Value Analysis

When we see password we have to check, whether password working or not.

Inputs:

- 1) OTP (sent to registered mobile)
- 2) New password
- 3) Confirm password

No of test cases in BVA Testing =  $4(n) + 1 = 13$  test cases

Conditions:

(1)  $\Rightarrow$  length = 6 digits

(2)  $\Rightarrow$  length = 8-12 digits

(3)  $\Rightarrow$  length = 8-12 digits

**Test matrix table:**

<b>Test case ID</b>	<b>OTP</b>	<b>New Password</b>	<b>Confirm Password</b>	<b>Expected Output</b>
TC01	123456	Abcdefghij	Abcdefghij	Password changed successfully
TC02	1234567	Abcdefghij	Abcdefghij	Invalid OTP
TC03	123	Abcdefghij	Abcdefghij	Invalid OTP
TC04	12345	Abcdefghij	Abcdefghij	Invalid OTP
TC05	123456	Abcdefgh	Abcdefghij	Password changed successfully
TC06	123456	Abcdefg	Abcdefghij	Invalid password length
TC07	123456	Abcdefghijklm	Abcdefghij	Invalid password length

TC08	123456	Abcdefghijkmi	Abcdefghij	Invalid password length
TC09	123456	Abcdefghij	Abcdefgh	Password changed successfully
TC10	123456	Abcdefghij	Abcdefg	Invalid password length
TC11	123456	Abcdefghij	Abcdefghijki	Invalid password length
TC12	123456	Abcdefghij	Ilbciiefghijki	Invalid password length
TC13	123456	Abcdefghij	Abcdefghij	Password changed successfully

### Test Execution:

Test cases Planned	Test cases Executed	Test cases Passed	Test cases Failed
20	13	4	9

## 5. JUNIT TESTING

JUnit is a widely used unit testing framework for Java programming language. It provides a set of libraries and annotations that help developers write and execute automated tests for their code. JUnit tests are designed to be simple, fast, and reliable, and they can be run automatically as part of a continuous integration (CI) pipeline to ensure that changes to the code do not introduce new bugs or regressions.

Here is a possible description of JUnit testing:

JUnit testing is a software testing practice that involves writing automated test cases for individual units of code, such as methods or functions. JUnit tests are typically written by developers themselves, and they are used to verify that the code behaves as expected and meets the requirements of the system. The JUnit framework provides a set of annotations, assertions, and test runners that help developers write and execute these tests.

To write a JUnit test, a developer first creates a test class that contains one or more test methods. Each test method is designed to test a specific unit of code, and it typically includes one or more assertions that verify the expected output or behavior of the code under test. The JUnit framework provides a set of built-in assertion methods, such as `assert Equals` and `assert True` that make it easy to write these assertions.

Once the test methods have been written, the developer can use a JUnit test runner to execute the tests and generate a report of the results. The JUnit test runner provides several options for executing tests, such as running all tests in a test

class or running a specific subset of tests. It also provides hooks for setting up and tearing down test fixtures, such as initializing database connections or creating mock objects.

The benefits of JUnit testing include increased confidence in the quality of the code, faster and more efficient debugging, and better overall code design. By writing automated tests that verify the behaviour of individual units of code, developers can catch bugs and regressions earlier in the development process, which can save time and money in the long run. Additionally, by designing code that is testable, developers can create code that is easier to maintain, refactor, and extend over time.

### **How to install Junit:**

JUnit testing in IntelliJ is a straightforward process. Here's a basic guide to get you started:

#### **1. Create a New Project**

First, you'll need to create a new Java project or open an existing one in IntelliJ IDEA.

#### **2. Add JUnit Dependency**

Make sure your project has JUnit added as a dependency. If you're using Maven, you can add it to your `pom.xml`:

#### **3. Create a Test Class**

Right-click on your source folder (like `src/main/java`) -> New -> Java Class.

#### **4. Write Test Methods**

Inside your test class, write methods to test your actual classes.

## 5. Run the Tests

To run a single test method, click on the small green triangle next to the test method name.

To run all tests in the project, right-click on the project folder and select "Run All Tests".

## 6. View Results

You'll see the test results in the "Run" tool window at the bottom. Green means the test passed, red means it failed, and yellow means it was ignored.

### Additional Tips:

IntelliJ usually recognizes JUnit tests automatically, so you'll see options to run or debug tests in the editor gutter next to your test methods.

You can also use annotations like `@Before` and `@After` to set up and tear down resources before and after tests.

IntelliJ has shortcuts like `Ctrl + Shift + T` (Cmd + Shift + T on macOS) to quickly create a new test or navigate between test and class files.

## **J-Unit Code:**

### **Login module:**

```
package com.mycompany.shoppersstop;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class LoginTest{

    public LoginTest() {

    }

    @org.junit.jupiter.api.BeforeAll
    public static void setUpClass() throws Exception {

    }

    @org.junit.jupiter.api.AfterAll
    public static void tearDownClass() throws Exception {

    }

    @org.junit.jupiter.api.BeforeEach public void setUp()
    throws Exception {

    }

    @org.junit.jupiter.api.AfterEach
    public void tearDown() throws Exception {
```



```
}  
/**  
 * Test of setUsername method, of class Login.  
 */ @org.junit.jupiter.api.Test  
public void testSetUsername(){  
    System.out.println("setUsername");  
    String un = "";  
    Login instance = new Login();  
    instance.setUsername(un);  
    // TODO review the generated test code and remove the  
    default call to fail.  
    //fail("The test case is a prototype.");  
}  
/**  
 * Test of setPassword method, of class Login.  
 */ @org.junit.jupiter.api.Test  
public void testSetPassword() {  
    System.out.println("setPassword");  
    String p = "";  
    Login instance = new Login();  
    instance.setPassword(p);  
    // TODO review the generated test code and remove the  
    default call to fail.
```

```
//fail("The test case is a prototype.");  
}  
/**  
 * Test of getUsername method, of class Login.  
 */ @org.junit.jupiter.api.Test  
public void testGetUsername() {  
    System.out.println("getUsername");  
    Login instance = new Login();  
    String expResult = "Prasanth1";  
    String result = instance.getUsername();  
    assertEquals(expResult, result);  
  
    // TODO review the generated test code and remove the  
    default call to fail.  
  
    //fail("The test case is a prototype.");  
}  
/**  
 * Test of getPassword method, of class Login.  
 */ @org.junit.jupiter.api.Test  
public void testGetPassword() {  
    System.out.println("getPassword");  
    Login instance = new Login();  
    String expResult = "Abc@123";  
    String result = instance.getPassword();
```

```
    assertEquals(expResult, result);

    // TODO review the generated test code and remove the
    default call to fail.

    //fail("The test case is a prototype.");
}

/**
 * Test of checkCredentials method, of class Login.
 */
@org.junit.jupiter.api.Test
public void testCheckCredentials() {
    System.out.println("checkCredentials");
    String un = "Prasanth1";
    String p = "Abc@123";
    Login instance = new Login();
    boolean expResult = true;
    boolean result = instance.checkCredentials(un, p);
    assertEquals(expResult, result);

    // TODO review the generated test code and remove the
    default call to fail.

    //fail("The test case is a prototype.");
}
}
```



## Payment Module:

```
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

/**
 *
 * @author pras
 */
public class PaymentTest {
    public PaymentTest() {
    }
}
```

@BeforeAll

```
public static void setUpClass() {  
}
```

@AfterAll

```
public static void tearDownClass() {  
}
```

@BeforeEach

```
public void setUp() {  
}
```

@AfterEach

```
public void tearDown() {  
}
```

/\*\*

\* Test of validateUser method, of class Payment.

\*/ @Test

```
public void testValidateUser() {  
    System.out.println("validateUser");  
    String un = "Prasanth1";  
    String p = "123";
```

```
Payment instance = new Payment();
boolean expResult = true;
boolean result = instance.validateUser(un, p);
assertEquals(expResult, result);

// TODO review the generated test code and remove the
default call to fail.

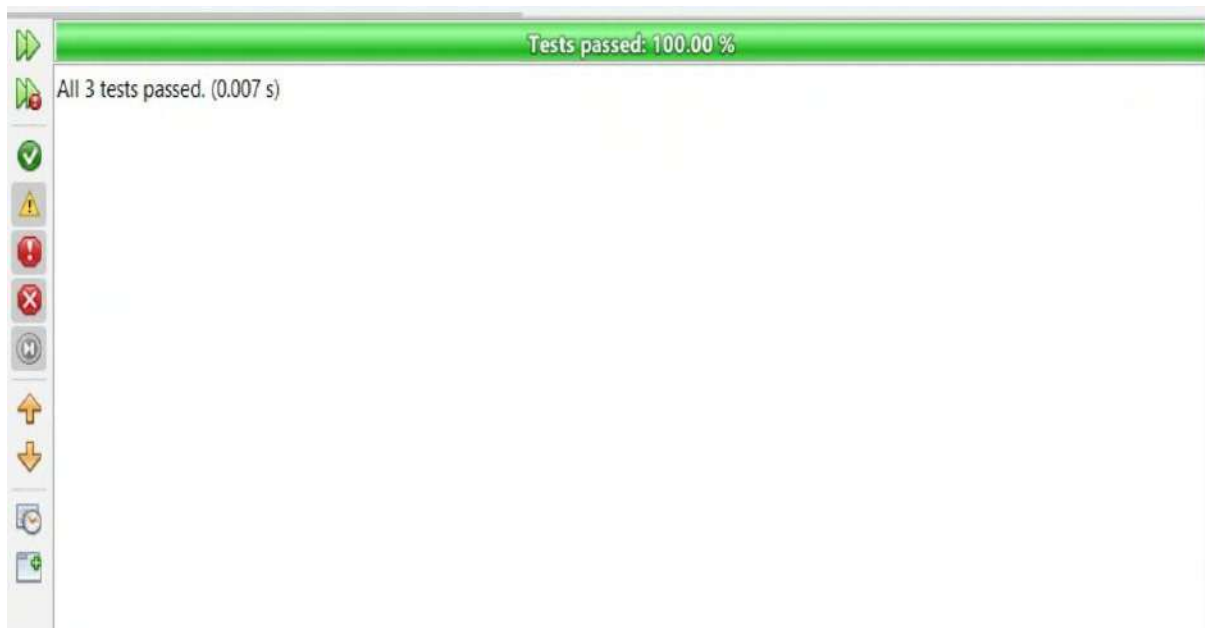
//fail("The test case is a prototype.");
}

/**
 * Test of selectItem method, of class Payment.
 */
@Test
public void testSelectItem() {
    System.out.println("selectItem");
    String item = "Bag";
    Payment instance = new Payment();
    instance.selectItem(item);

    // TODO review the generated test code and remove the
    default call to fail.

    // fail("The test case is a prototype.");
}
```

```
/**  
 * Test of generateBill method, of class Payment.  
 */ @Test  
public void testGenerateBill() {  
    System.out.println("generateBill");  
    String un = "Prasanth1";  
    String p = "Abc@123";  
    String item = "Bag";  
    Payment instance = new Payment();  
    instance.generateBill(un, p, item);  
    // TODO review the generated test code and remove the  
    default call to fail.  
    //fail("The test case is a prototype.");  
}  
}
```



## For failed test cases:



## TestRigor Test Automation Tool:

TestRigor automates the creation and execution of end-to-end tests, primarily for web and mobile applications. Its user base consists of quality assurance professionals and testers who seek efficiency without sacrificing depth in testing. In this testRigor review, you'll learn the tool benefits users by enabling them to write tests in plain English, significantly reducing the learning curve and accelerating test creation. Moreover, this testRigor review will address the pain point of intricate test script maintenance by allowing users to focus on test scenarios rather than the underlying code. Its best features include natural language processing for test creation, robust integration options, and advanced analytics for test results.

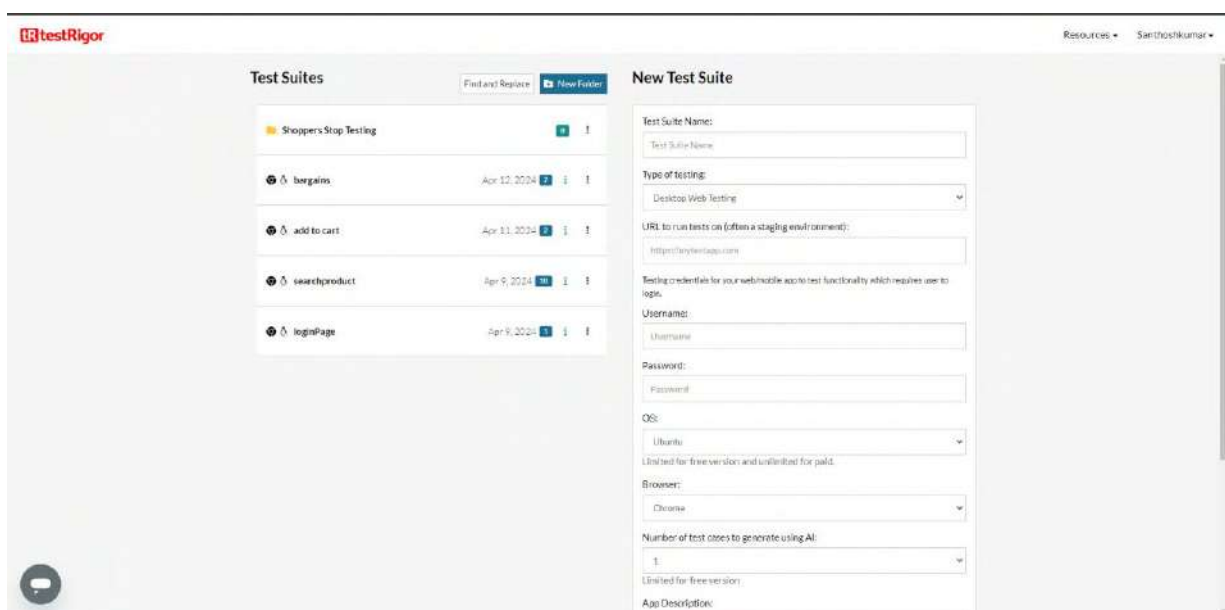




testRigor serves well for organizations that prioritize rapid test creation without deep coding expertise. The most loyal customers are those who value efficiency in test automation, such as agile teams in small to mid-sized tech companies. It's also well-suited for teams that have to manage a significant amount of UI testing across web and mobile applications.

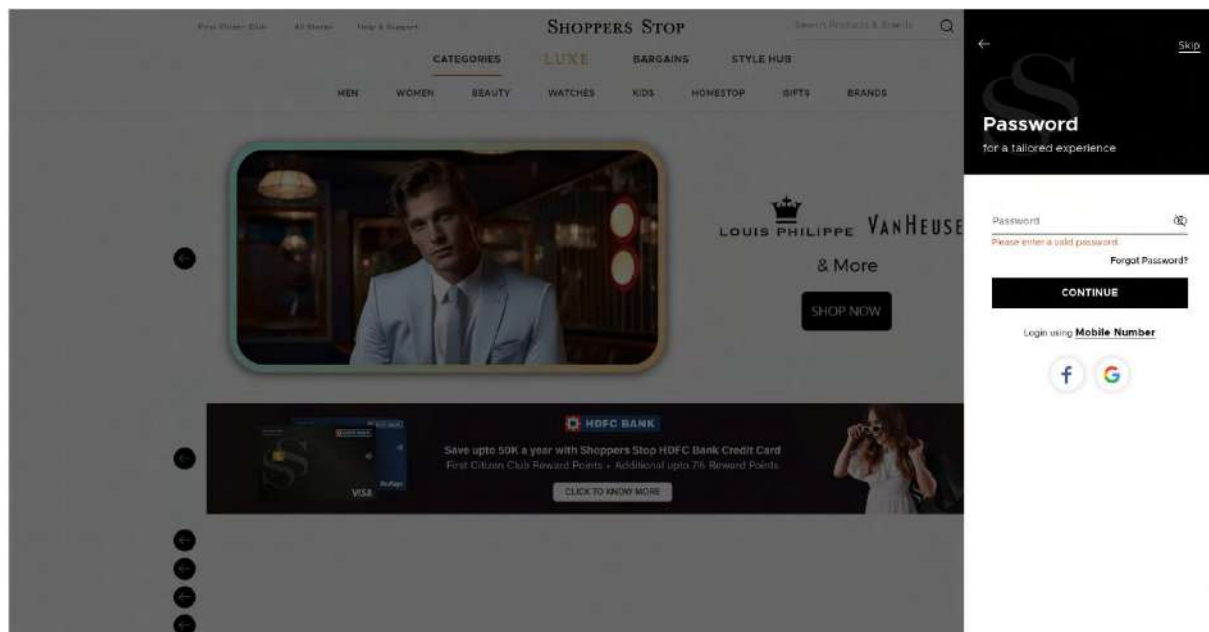
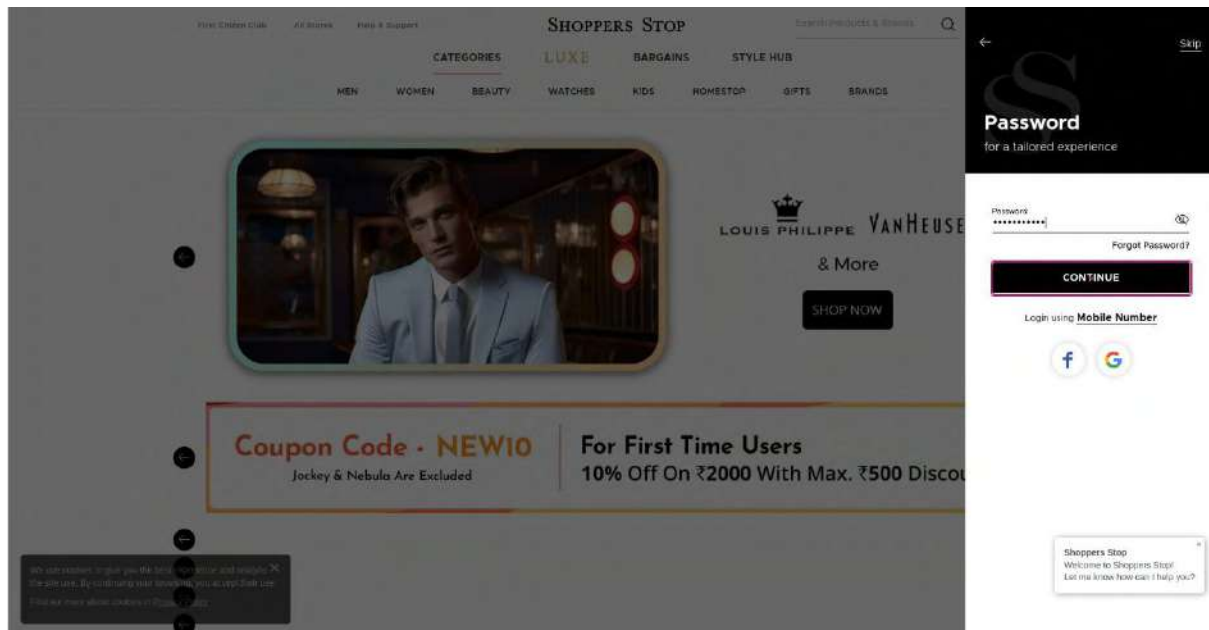
Tests that can be performed by TestRigor are:

1. Web Application
2. Mobile Application
3. API
4. Desktop Window application
5. Email



## Test Cases for log in:

Test Case ID	Test Case Description	Input Value	Expected Result	Actual Result	Test Case Status
TC001	Login Using Email	1)password 2)user_name	The account is now logged in	Login successful	Valid Test case Passed
TC002	Logging using Email	1)user_name 2)-----	Wrong password	Enter a password Login failed	Invalid Test case Passed
TC003	Login Using Email	1)----- 2)password	user didn't enter an Email	Enter an Email Id login failed	Invalid Test case Failed
TC004	Login Using Email	1)incorrect username 2)password	User didn't enter an valid user_name	Enter an valid username Login failed	Invalid Test Failed Passed
TC005	Login Using Email	1)user_name 2>Password (long)	The password Is too long	Enter a Password under 6 characters Login failed	Invalid Test case Passed
TC006	Login Using Email	1)username 2)password (short)	Password must be 6 characters	Password is too short enter an valid password Login failed	Invalid Test case Passed
TC007	Login Using Email	1)username (NON-EXISTING) 2)password	Invalid login credentials	Invalid login credentials Try again Login failed	Invalid Test case Passed
TC008	Login Using Email	1)username 2)password (wrong)	Invalid login credentials	Invalid login Credentials Try again Login failed	Invalid Test case Passed



## Test Cases for Search Product:

Test Case ID	Test Case Description	Input Value	Expected Result	Actual Result	Test Case Status
TC001	Search Product	watch	User can see watches	User can see watches	Valid Test case Passed
TC002	Search a specific Product	Timex (men)	User can see timex(men) watch	User can see timex(men) watch	Valid Test case Passed
TC003	Search product	----- -	user didn't enter valid input	No Results found	Invalid Test case Passed
TC004	Search product	abcd	User can't find the product	We didn't find what you were looking for	Invalid Test case Passed
TC005	Search product with offers	Black shirt 50% OFF	User can see watchs with 50% OFF	User can see watchs with 50% OFF	Valid Test case Passed
TC006	Search product with offers	Black shirt 2% OFF	User can't see watchs with 2% OFF	No Results found	Invalid Test case Passed
TC007	Search product with brand	American Eagle Shirt	User can see American Eagle Brand shirt	User can see American Eagle Brand shirt	Valid Test case Passed
TC008	Search product with brand	Russian Shirt	User can'tsee Russian Brand shirt	No Results found	Invalid Test case Passed

TC009	Search with price range	Shirt price 2000 to 5000	User can see shirt with the price range	User can see shirt with the price range	Valid Test case Passed
TC010	Search with price range	Shirt price 1 to 2	User can't see shirt with the price range	We didn't find what you were looking for. Please try using a different search term.	Invalid Test case Passed

## Execution of test cases:

The screenshot shows the testRigor web interface. On the left is a sidebar with navigation options: All Test Suites, Test Suite Details, Test Cases (11), My requests, Behavior-Driven Test Creation, Last Run (Failed), Errors (1), Reports, Tree View, Captured Data, All Runs, Historical Reports, Live Mode, and Suite. The main area displays a list of test cases:

- Name:** Product Review and Rating Submission Test  
**Steps:** Login. Enter "Samsung Galaxy S22" into "Search Products & Brands". Click "nars Powermatte Lip Pigment Liquid Lipstick". Click "Submit".  
**Labels:** +  
**Link Data Set:** Failed 43 secs
- Name:** User Profile and Order History Accuracy Test  
**Steps:** Login. Click "help & support". Click "TRACK ORDER". Enter "123456789" into "Enter order number".  
**Labels:** +  
**Generate corner cases:** Link Data Set: Passed 2 min 57 secs
- Name:** Order Confirmation and Email Notification Check  
**Steps:** Login. Enter "Test Product" into "Search Products & Brands". Click "nars Powermatte Lipstick ₹ 3000". Click "Add to bag" below the "Return or exchange...".  
**Labels:** +  
**Generate corner cases:** Link Data Set: Passed 2 min 39 secs
- Name:** Checkout Process and Payment Gateway Integration Test  
**Steps:** Login. Enter "iPhone 13" into "Search Products & Brands". Click "nars Powermatte Lip Pigment Liquid Lipstick". Click "PENTS...".  
**Labels:** +  
**Generate corner cases:** Link Data Set: Passed 2 min 48 secs
- Name:** Cart Update and Validation Test  
**Steps:** Login. Enter "TestProduct123" into "Search Products & Brands". Click "nars Radiant Creamy Color Corrector ₹ 2800". Click "Add to bag" below the "Return or exchange...".  
**Labels:** +  
**Generate corner cases:** Link Data Set: Passed 2 min 6 secs
- Name:** Product Details Page Accuracy Verification  
**Steps:** Enter "Nike Shoes" into "Search Products & Brands". Click "nars Powermatte Lip Pigment Liquid Lipstick". Click "Product Details". Check if page contains...  
**Labels:** +  
**Generate corner cases:** Link Data Set: Passed 1 min 56 secs
- Name:** Add to Cart Functionality Test  
**Steps:** Enter "iPhone 13" into "Search Products & Brands". Click "nars Powermatte Lipstick ₹ 3000". Click "Add to bag" below the "Return or exchange not applica...".  
**Labels:** +  
**Generate corner cases:** Link Data Set: Passed 1 min 1 secs
- Name:** Product Search Functionality Test  
**Steps:** Enter "Nike Shoes" into "Search Products & Brands". Click "Submit".  
**Labels:** +  
**Generate corner cases:** Link Data Set: Passed 50 secs

This section continues the list of test cases from the previous screenshot:

- Name:** Add to Cart Functionality Test  
**Steps:** Enter "iPhone 13" into "Search Products & Brands". Click "nars Powermatte Lipstick ₹ 3000". Click "Add to bag" below the "Return or exchange not applica...".  
**Labels:** +  
**Generate corner cases:** Link Data Set: Passed 1 min 2 secs
- Name:** Product Search Functionality Test  
**Steps:** Enter "Nike Shoes" into "Search Products & Brands". Click "Submit".  
**Labels:** +  
**Generate corner cases:** Link Data Set: Passed 50 secs
- Name:** User Login Functionality Check  
**Steps:** Login.  
**Labels:** +  
**Generate corner cases:** Link Data Set: Passed 1 min 2 secs

At the bottom, it shows "11 results | Page 1 of 2" with navigation arrows.

## Test Cases:

### TC01:

### Login functionality:

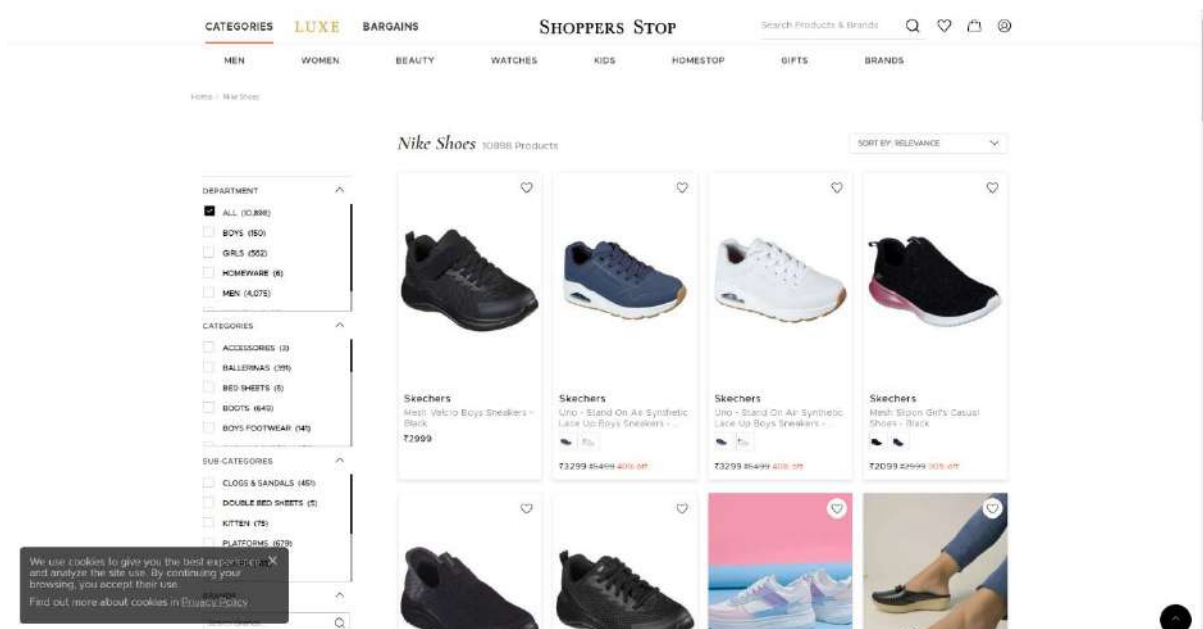
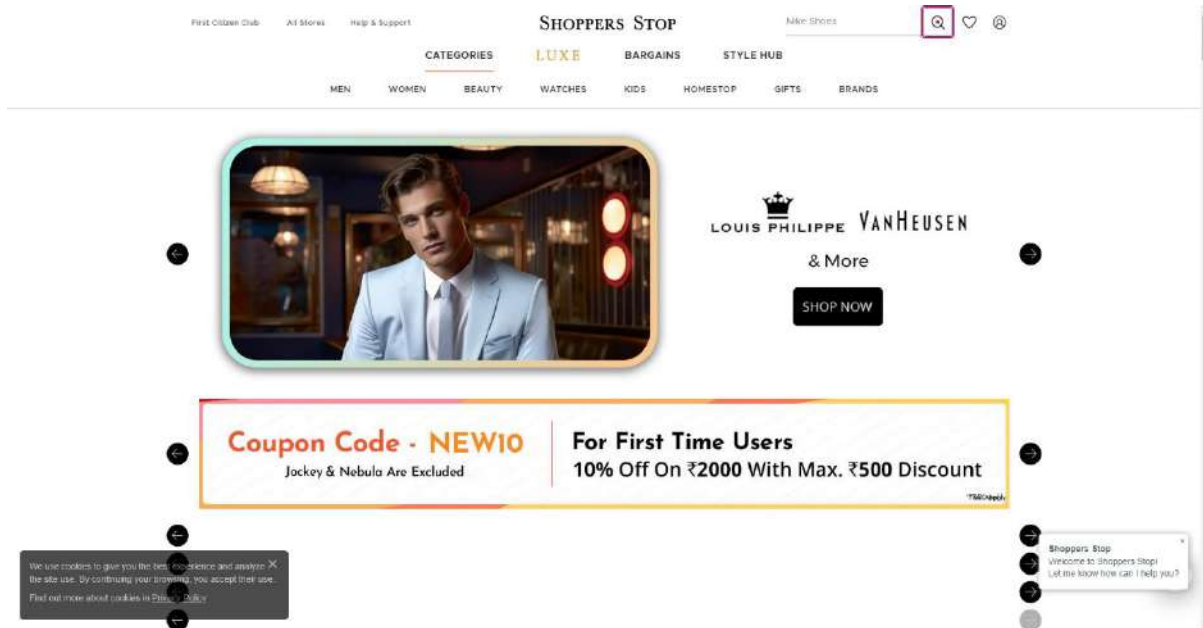
The screenshot shows the testRigor interface for a test case named "User Login Functionality Check". The test steps are "Login". The test was created on 04/13/2024 at 11:43 by Madhu Prasanth and triggered on the same date at 11:43. The UUID is 44d8ed2-b7f4-4f76-a148-e02edc5b60f9. The test status is "Failed". The test results show two screenshots of the login page. The first screenshot shows the login form with a "Collapse" button and a "Show extra info" button. The second screenshot shows the login form with a "More details..." button. The test results also show a "15s 79ms" duration for the first step and a "15s 93ms" duration for the second step.

### TC02:

### Search Nike shoe:

The screenshot shows the testRigor interface for a test case named "Product Search Functionality Test". The test steps are "Enter 'Nike Shoes' into 'Search Products & Brands'. Click 'Submit'". The test was created on 04/13/2024 at 11:43 by Madhu Prasanth and triggered on the same date at 11:43. The UUID is 6db95d45-b621-4f57-92ef-11b1e0233e70. The test status is "Failed". The test results show five screenshots of the search results page. The first screenshot shows the search bar and the "More details..." button. The second screenshot shows the search results with a "Description: enter 'Nike Shoes' into 'Search Products & Brands'" and a "15s 167ms" duration. The third screenshot shows the search results with a "Description: click 'Submit'" and a "15s 167ms" duration. The fourth screenshot shows the search results with a "Description: click 'Submit'" and a "15s 167ms" duration. The fifth screenshot shows the search results with a "Description: click 'Submit'" and a "15s 167ms" duration.

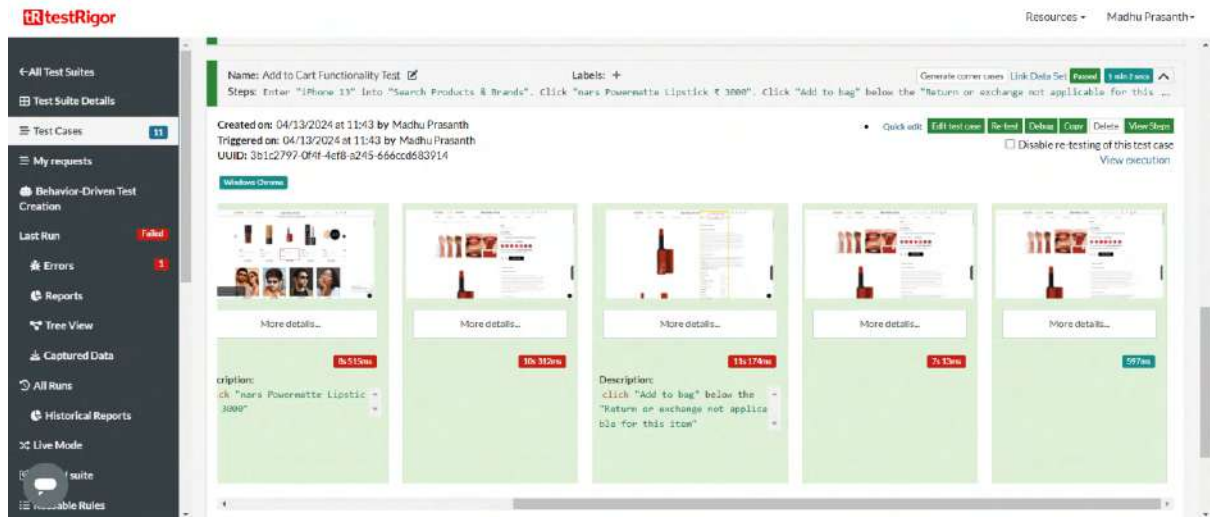
# Result:



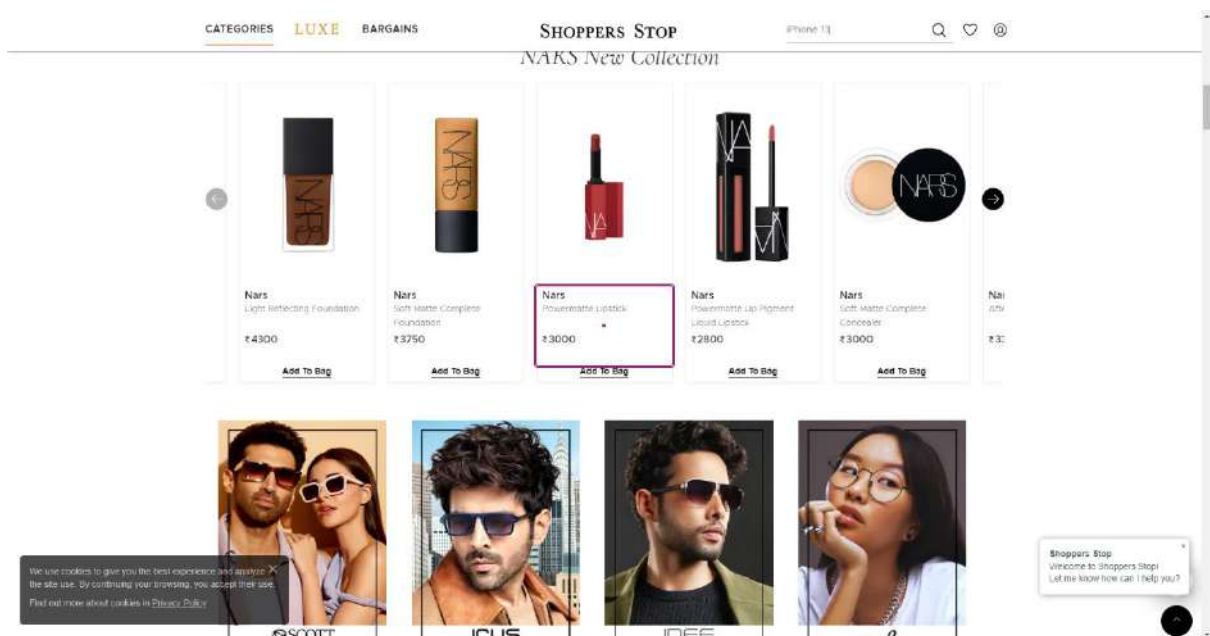


TC03:

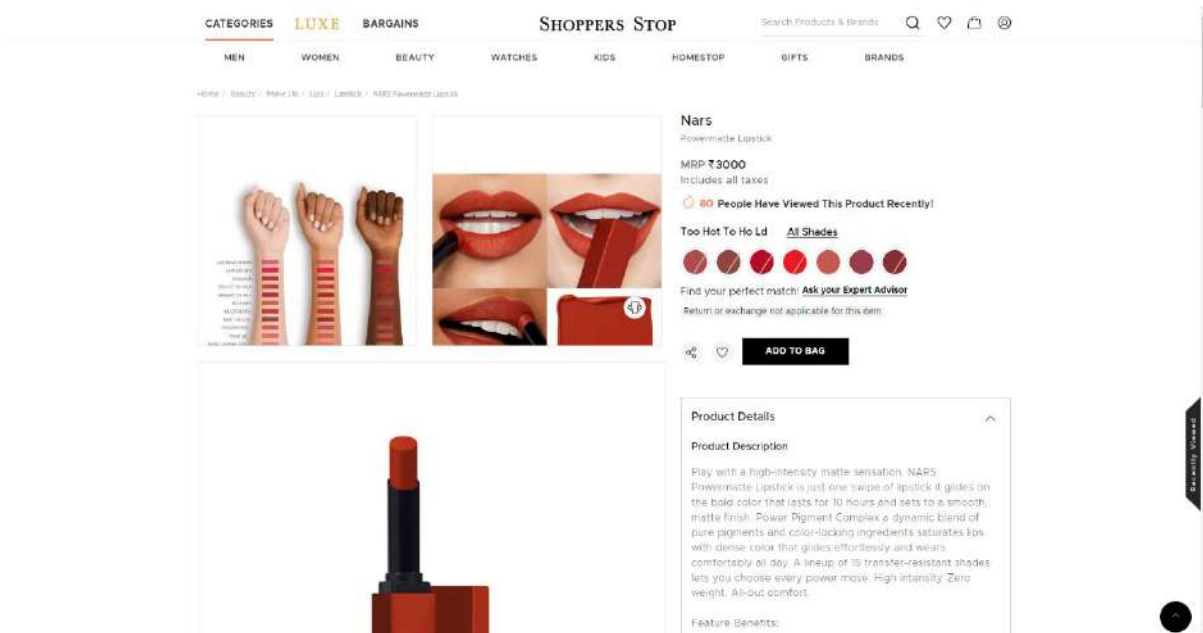
Add to cart functionality:



Result:

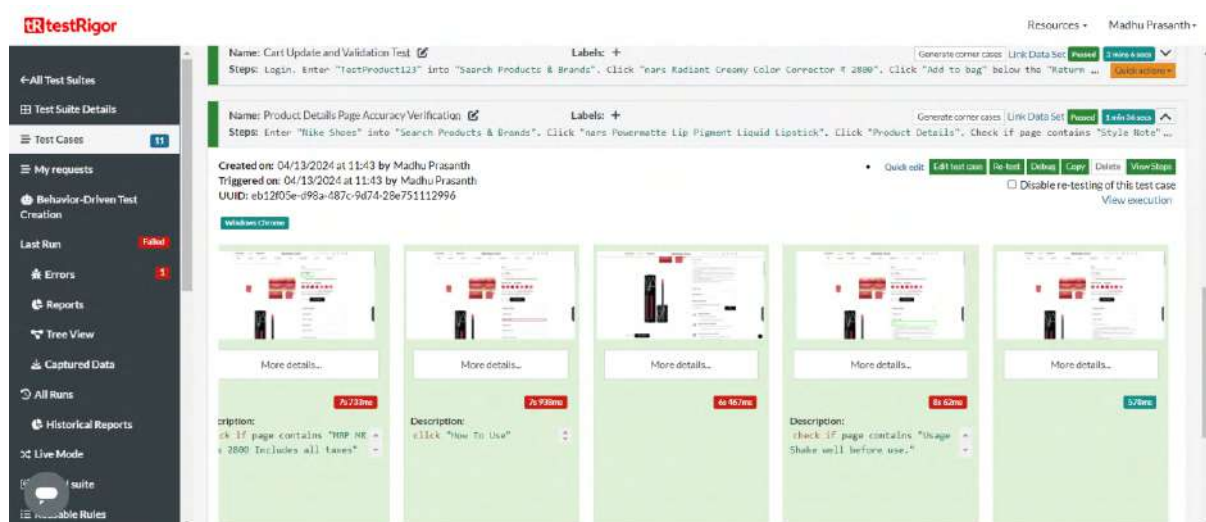




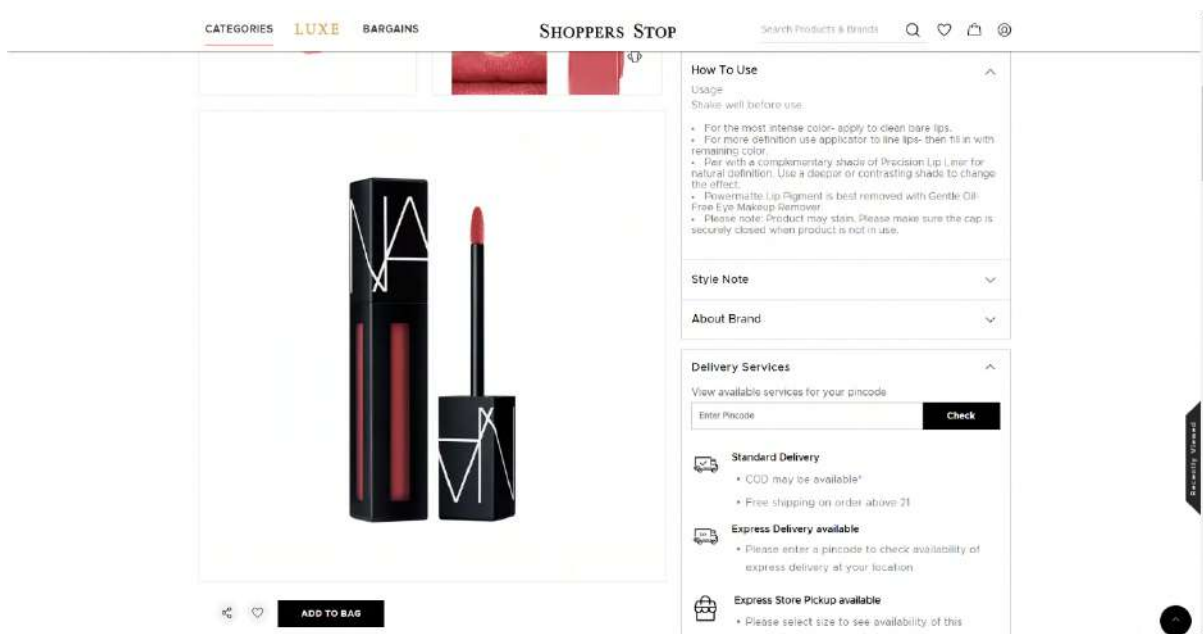
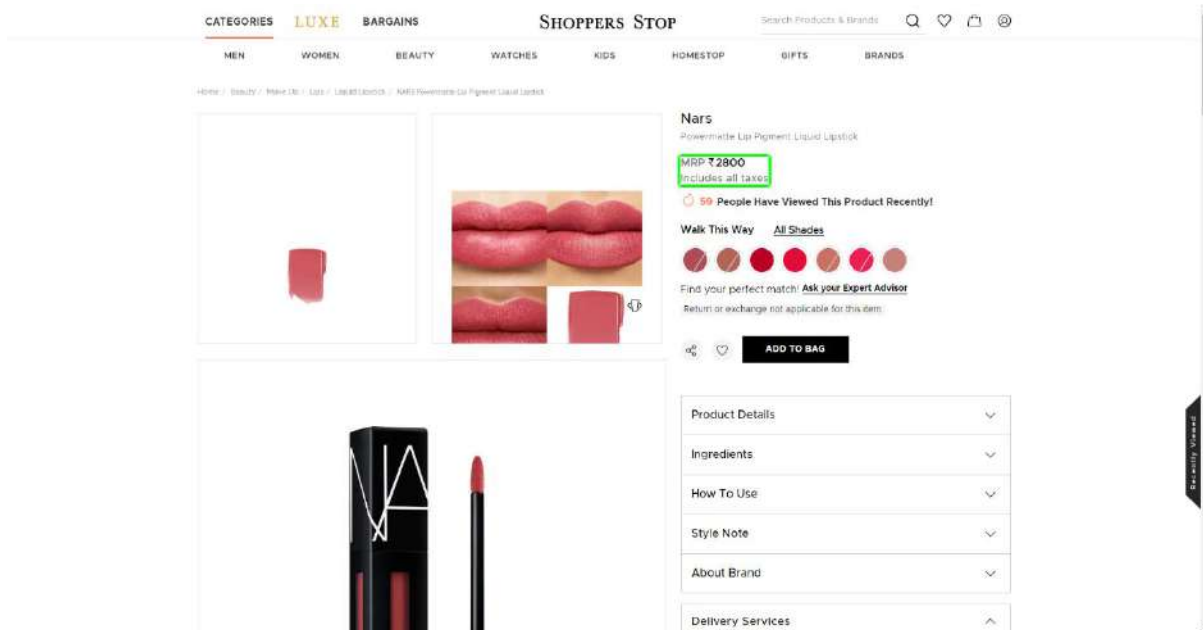


## TC04:

### Product page details accuracy:

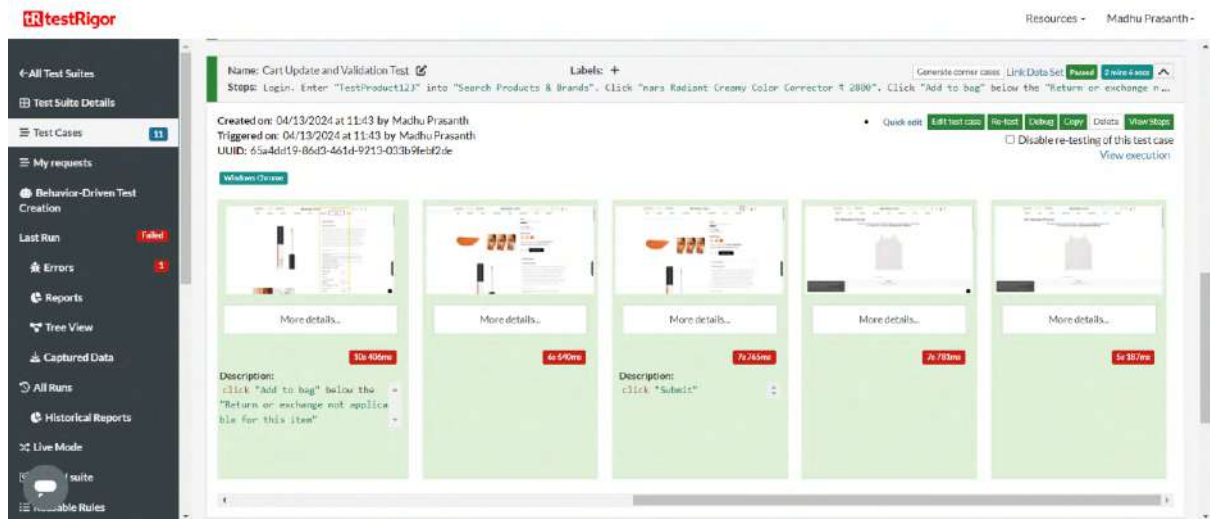


# Result:

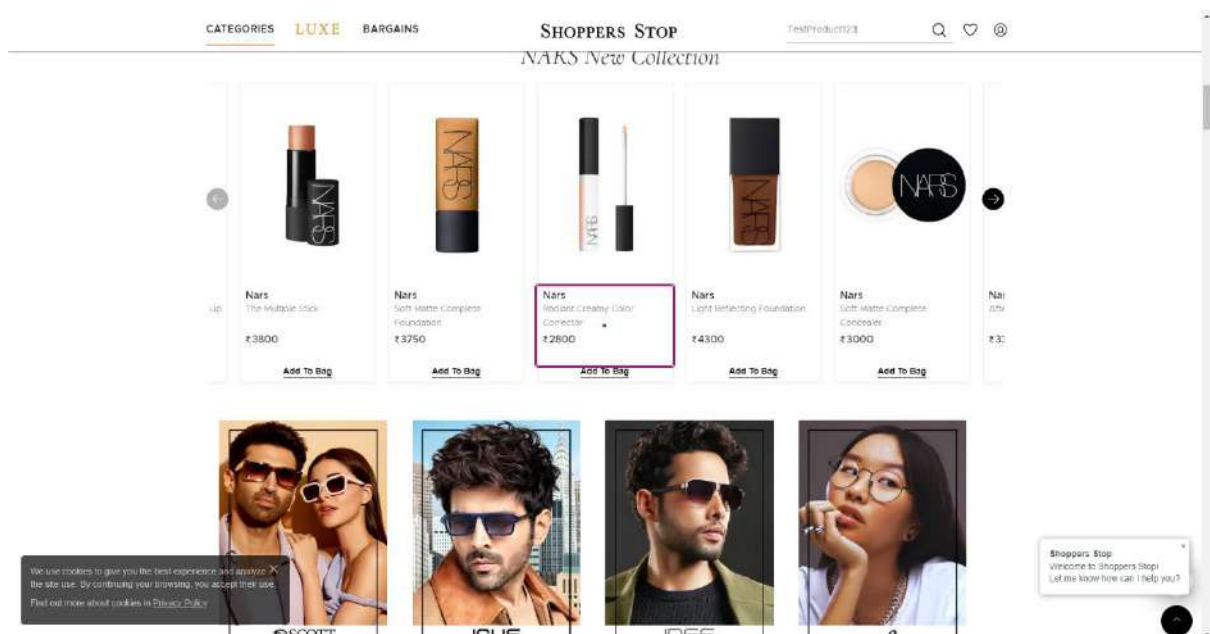


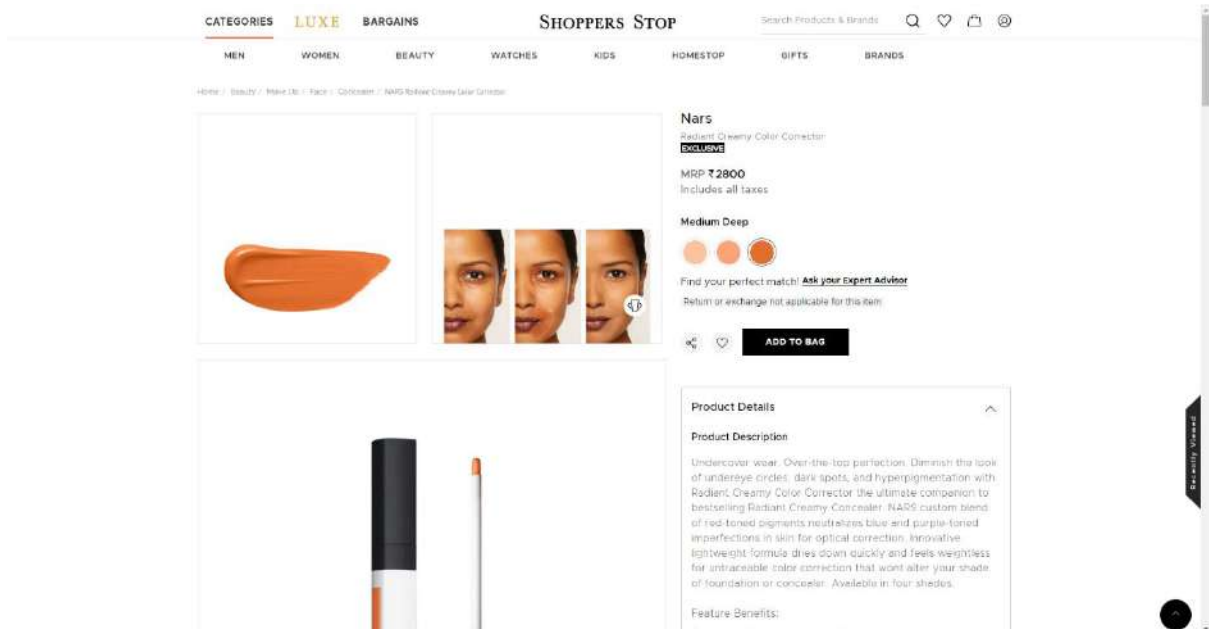
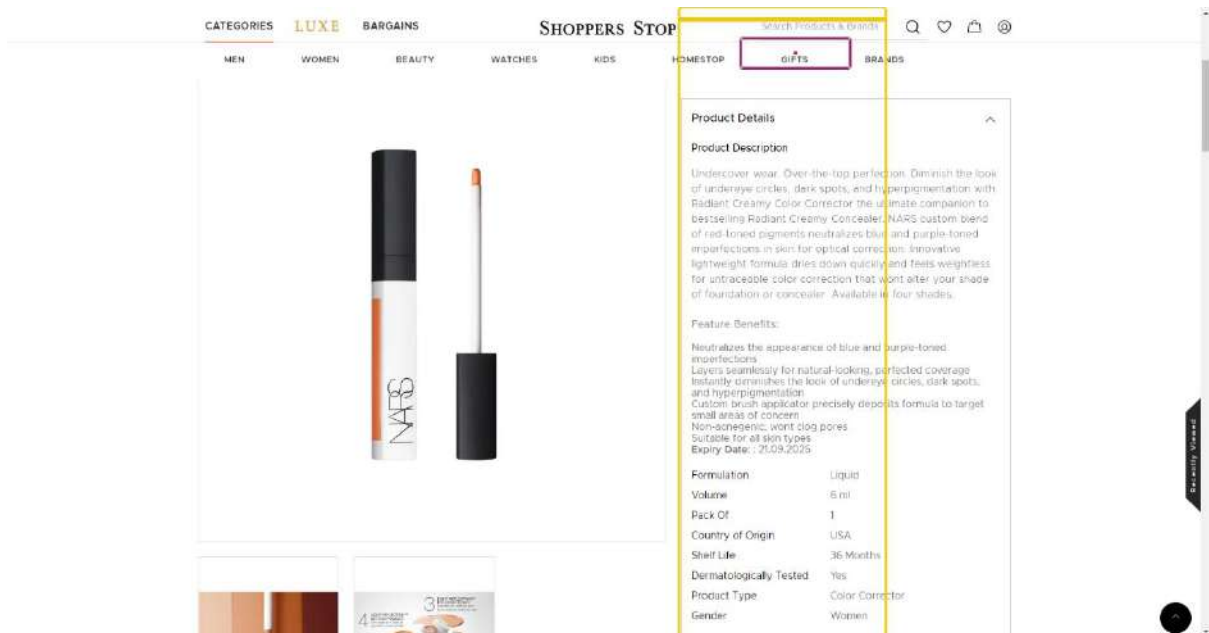
## TC05:

### Cart update and validation test:



## Result:





## TC06:

## User and order history:

The screenshot displays the TestRigor web interface. On the left is a dark sidebar with navigation options: All Test Suites, Test Suite Details, Test Cases (11), My requests, Behavior-Driven Test Creation, Last Run (Failed 3), Errors, Reports, Tree View, Captured Data, All Runs, Historical Reports, Live Mode, Suite, and Repeatable Rules. The main area shows a test case titled 'User Profile and Order History Accuracy Test' with labels '+'. The steps are: 'Login, click "Help & Support", click "TRACK ORDER", Enter "123456789" into "Enter order number"'. It was created on 04/13/2024 at 11:43 by Madhu Prasanth, triggered on the same date at 11:43, with a UUID of 322cbb7-abc5-4f4f-ac84-30b3ca55e350. Below the steps are five screenshot thumbnails of the application under test, each with a description and a status (e.g., 'Description: click "Help & Support"', 'Status: 100.00%').

## Result:

The screenshot shows the Shoppers Stop website's 'TRACK ORDER' page. The header includes 'CATEGORIES', 'LUXE', 'BARGAINS', and 'SHOPPERS STOP'. Below the header are navigation links: MEN, WOMEN, BEAUTY, WATCHES, KIDS, HOMESTOP, GIFTS, and BRANDS. The main content area has a 'TRACK ORDER' heading and a form to track an order by entering an email address and an order number (123456789). A 'Submit' button is at the bottom of the form. To the right, a 'Need help?' section lists links: Help/FAQs, Track order (highlighted), Size Guide, Buying Guide, How do I shop?, How do I pay?, and Find places we deliver. The footer contains various sections: 'Authentic Products', 'Express Store Pickup', 'Secure Payment', 'Easy return and Exchange', 'PAY SECURELY BY' (with logos for Visa, Mastercard, American Express, and PayPal), 'REACH US' (with contact information), 'DOWNLOAD THE APP ON' (with Google Play and App Store logos), 'VERIFIED BY' (with logos for Visa, Mastercard, American Express, and PayPal), 'FOLLOW US ON' (with social media icons), 'CUSTOMER' (with a cookie notice), 'STORE AND SITES' (with links for About Us, Contact Us, Corporate Site, Store Locator, and Gift Cards), 'POLICIES' (with links for Terms of Use, Privacy, Delivery Policy, and Exchanges & Returns), 'FIRST CITIZEN' (with links for First Citizen Club and First Citizen Card), and 'DELIGHTFUL PROGRAMS' (with links for Instant Gifting and Express Store Pick Up).

CATEGORIES

LUXE

BARGAINS

SHOPPERS STOP

Search Products & Brands

MEN

WOMEN

BEAUTY

WATCHES

KIDS

HOMESTOP

GIFTS

BRANDS

OUR ADDRESS

Shoppers Stop Limited,  
Umang Tower, 5th Floor,  
MindSPACE, Off Link Road,  
Mumbai (West), Maharashtra - 400 064

Submit Your Query

Fields are mandatory

First name\*

Last name\*

Your email address\*

Phone

Your mobile number\*

Order number

Captcha

Comments (Max 100 characters)

75597

0:00 / 0:00

Click to change

Clear Captcha

SUBMIT

Need Help!

For Customer Support & Queries:

Call us at Customer Care no. : 07949721777

Email us at [customerscare@shoppersstop.com](mailto:customerscare@shoppersstop.com)

(Operational: Tuesday: 08:00AM to 10:00PM)

We use cookies to give you the best experience and analyze site usage. By continuing your browsing, you accept their use.

Find out more about cookies in [Privacy Policy](#)

Shoppers Stop

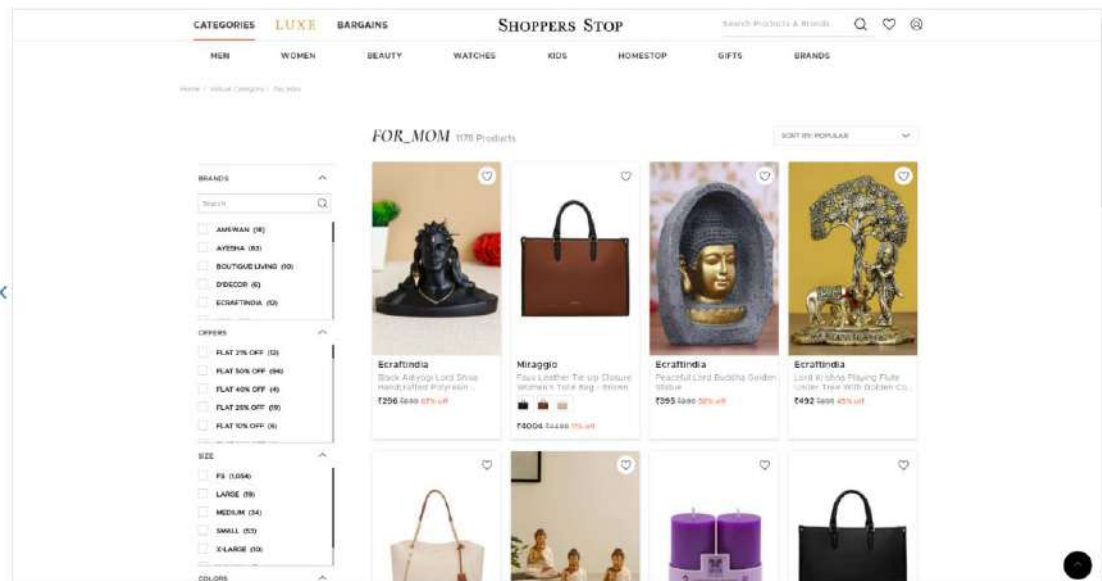
Welcome to Shoppers Stop!

Let us know how can I help you?

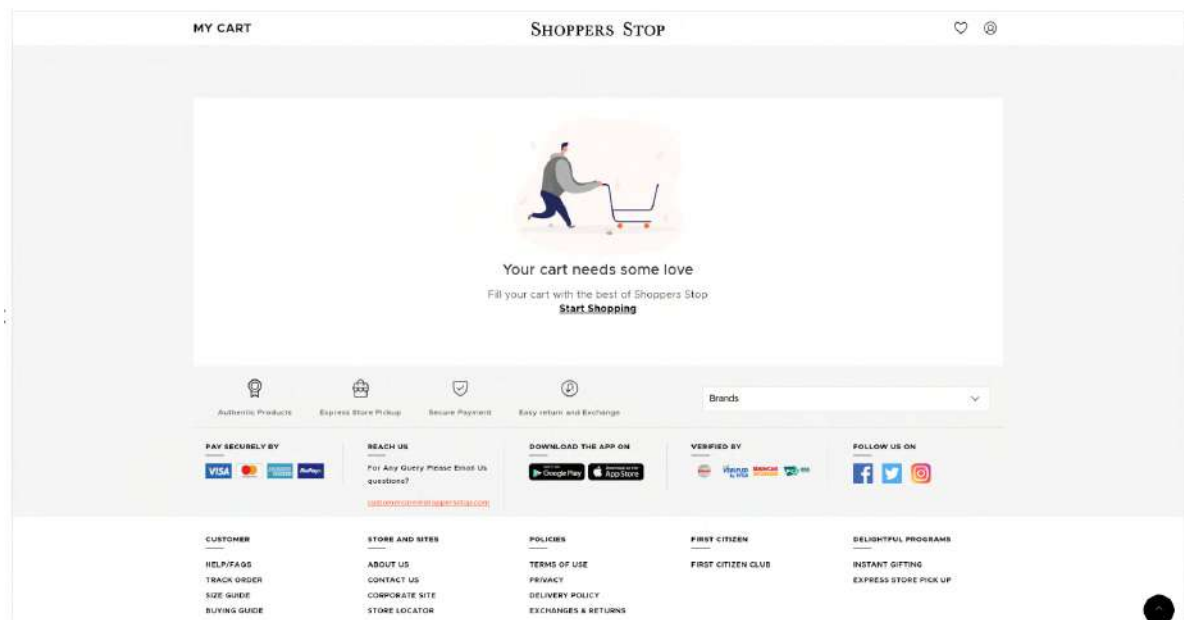
Showing products from 50-70% off:

The screenshot displays the Shoppers Stop website interface. At the top, there are navigation tabs for CATEGORIES, LUXE, and BARGAINS. Below these are sub-tabs for MEN, WOMEN, BEAUTY, WATCHES, KIDS, HOMESTOP, GIFTS, and BRANDS. A search bar is located on the right side of the header. The main content area is titled 'Private brands' and shows a grid of 8 product images. Each product image is accompanied by a discount tag indicating a percentage off (e.g., 50% OFF, 70% OFF). On the left side, there is a sidebar with filters for DEPARTMENT (ALL, BOYS, GIRLS, MEN, WOMEN), CATEGORIES (ACCESSORIES, CASUALS, SHIRTS, T-SHIRTS, JEANS, JACKETS, etc.), and SUB-CATEGORIES (BOTTOMWEAR, SHOES, etc.). A cookie consent banner is visible at the bottom left, and a 'Need Help?' section is at the bottom right.

## Gifts for mother:



## No product in cart:





## **How to use:**

TestRigor is a platform designed to automate software testing, making the testing process more efficient and reliable. Here's a general guide on how to perform testing using TestRigor:

### **1. Create Test Cases:**

Start by defining your test cases. These are the scenarios or steps that you want the system to execute and verify. TestRigor uses natural language processing to create test cases, so you can write them in plain English.

### **2. Record Test Scripts:**

TestRigor provides a browser extension that allows you to record your actions as you navigate through your application. This captures your interactions and generates test scripts automatically.

### **3. Edit Test Scripts (if necessary):**

Once the test scripts are generated, you may need to edit them to refine the test cases or make them more robust. This step ensures that your test cases accurately reflect the desired behavior of your application.

### **4. Run Tests:**

After creating and editing your test scripts, you can run them using Test Rigor's testing platform. TestRigor executes the test scripts in different browsers and environments to ensure compatibility.



## **5. Review Results:**

Once the tests are completed, you could review the results to see if any tests failed. TestRigor provides detailed reports and logs to help you identify issues and diagnose the root cause of failures.

## **6. Iterate and Retest:**

If any tests fail, you can iterate on your test cases, scripts, or application code to address the issues and then retest until all tests pass satisfactorily.

## **7. Continuous Integration:**

TestRigor can be integrated into your continuous integration/continuous deployment (CI/CD) pipeline to automate testing as part of your development workflow. This ensures that new code changes are thoroughly tested before being deployed to production.

By following these steps, you can effectively use TestRigor to automate your software testing process and ensure the quality and reliability of your applications.

## **Advantages:**

TestRigor offers several advantages as a testing tool:

### **1. Automated Testing:**

TestRigor automates the testing process, reducing the need for manual testing. Automated testing saves time and resources while increasing test coverage and consistency.

## **2. Natural Language Processing:**

TestRigor uses natural language processing (NLP) to create test cases, allowing testers to write tests in plain English. This makes test creation more accessible to non-technical team members and accelerates the test creation process.

## **3. Codeless Test Automation:**

TestRigor's codeless test automation approach eliminates the need for writing complex code to create and execute tests. This lowers the barrier to entry for testers and enables teams to focus on testing rather than programming.

## **4. Cross-Browser Testing:**

TestRigor supports cross-browser testing, allowing you to run tests across different web browsers and environments. This ensures that your application functions correctly for all users, regardless of their choice of browser.

## **5. Continuous Testing:**

TestRigor integrates seamlessly with continuous integration/continuous deployment (CI/CD) pipelines, enabling automated testing as part of your development workflow. Continuous testing helps catch bugs early in the development process and ensures that new code changes do not introduce regressions.

## **6. Fast Test Execution:**

TestRigor executes tests quickly, providing rapid feedback on the quality of your application. Fast test execution speeds up the testing cycle and enables faster delivery of software updates.

## **7. Comprehensive Reporting:**

TestRigor generates detailed reports and logs for test runs, allowing you to track test results and analyse any failures. Comprehensive reporting helps identify issues quickly and facilitates troubleshooting and debugging.

## **8. Scalability:**

TestRigor is scalable and can handle testing for projects of any size, from small applications to large enterprise systems. Whether you're testing a single feature or an entire application, TestRigor can accommodate your testing needs.

Overall, TestRigor offers a user-friendly and efficient testing solution that streamlines the testing process, improves test coverage, and enhances the quality of your software products.

## **Disadvantages:**

While TestRigor offers many advantages in terms of automating software testing, like any tool, it also has its limitations and potential disadvantages:

## **1. Limited Support for Complex Testing Scenarios:**

TestRigor's natural language processing may struggle with highly complex or specialized testing scenarios. Test cases that involve intricate business logic or interactions may not be easily captured or automated.

## **2. Dependency on User Interface (UI):**

TestRigor primarily focuses on UI testing, which means it may not be suitable for testing non-UI components such as APIs, databases, or backend services. This limitation could result in incomplete test coverage for certain types of applications.

## **3. Maintenance Overhead:**

Like any test automation tool, test scripts created with TestRigor require maintenance over time. Changes to the application's UI or functionality may necessitate updates to the test scripts, which can add to the overall maintenance overhead.

## **4. Learning Curve:**

While TestRigor aims to simplify the test automation process with its natural language approach, there may still be a learning curve for users who are new to the tool or to test automation in general. Training and familiarization may be required to effectively utilize all TestRigor's features.

## **5. Limited Customization Options:**

TestRigor's approach to test automation may not provide the level of customization needed for certain testing scenarios. Users may find themselves constrained by the tool's predefined capabilities, especially when attempting to implement highly specific or advanced testing techniques.

## **6. Cost:**

Depending on the scale of your testing needs and the features you require, TestRigor's pricing may be a disadvantage for some organizations. The cost of the tool may be prohibitive for smaller teams or projects with limited budgets.

## **7. Vendor Lock-In:**

Using a proprietary test automation tool like TestRigor may lead to vendor lock-in. If your testing processes become heavily dependent on TestRigor, transitioning to another tool or platform in the future could be challenging and disruptive.

Overall, while TestRigor offers automation capabilities that can streamline the testing process, it's essential to consider these potential disadvantages and evaluate whether the tool aligns with your specific testing requirements and organizational goals.

## White Box Testing:

## Static Data Flow Testing:

Module used: Sign-in module

Code:

```
1 private void IN_B_SIGNINActionPerformed(java.awt.event.ActionEvent evt) {  
2     // TODO add your handling code here:  
3     String a,b;  
4     a=IN_USERNAME.getText();  
5     b=IN_PASSWORD.getText();  
6     if(a.isEmpty() && b.isEmpty())  
7     {  
8         jLabel8.setText(".. Fill the username and password");  
9     }  
10  
11     if(!a.isEmpty() && !b.isEmpty())  
12     {  
13         try  
14         {  
15             Class.forName("com.mysql.cj.jdbc.Driver");  
16             Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/stadium","root","");  
17             String s="select email,pass,role from signup where email=? and pass=?";  
18             PreparedStatement pat=con.prepareStatement(s);  
19             pat.setString(1,a);  
20             pat.setString(2,b);  
21             ResultSet rs=pat.executeQuery();  
22             if(rs.next())  
23             {  
24                 String role =rs.getString("role");  
25                 switch (role)  
26                 {  
27                     case "admin":  
28                     {  
29                         admin in =new admin(a);  
30                         in.show();  
31                         dispose();  
32                         break;  
33                     }  
34                     case "customer":  
35                     {  
36                         customer in =new customer(a);  
37                         in.show();  
38                         dispose();  
39                         break;  
40                     }  
41                     case "high":  
42                     {  
43                         high in =new high(a);  
44                         in.show();  
45                         dispose();  
46                         break;  
47                     }  
48                 }  
49             }  
50             else  
51             {  
52                 jLabel8.setText(" Invalid Username or Password");  
53             }  
54         }  
55         catch(Exception e)  
56         {  
57             System.out.println(e);  
58         }  
59     }  
60     else  
61     {  
62         jLabel8.setText(" Invalid Username or Password");  
63     }  
64 }
```

Variables found are:

a, b, s, pat, rs, in, in, in

**For variable a:**

Pattern	Line Number	Explanation
~d	4	Normal Case, Allowed
du	4-6	Normal Case, Allowed
uu	6-11,19-20	Normal Case, Allowed
uk	43-64	Normal Case, Allowed
k~	64	Normal Case, Allowed

**For variable b:**

Pattern	Line Number	Explanation
~d	5	Normal Case, Allowed
du	5-6	Normal Case, Allowed
uu	6-11,19-20	Normal Case, Allowed
uk	20-64	Normal Case, Allowed
k~	64	Normal Case, Allowed

**For variable s:**

Pattern	Line Number	Explanation
~d	17	Normal Case, Allowed
du	17-18	Normal Case, Allowed
uu	-	Normal Case, Allowed
uk	18-64	Normal Case, Allowed
k~	64	Normal Case, Allowed

**For variable pat:**

Pattern	Line Number	Explanation
~d	18	Normal Case, Allowed
du	18-19	Normal Case, Allowed
uu	19-20, 20-21	Normal Case, Allowed
uk	21-64	Normal Case, Allowed
k~	64	Normal Case, Allowed

**For variable rs:**

Pattern	Line Number	Explanation
~d	21	Normal Case, Allowed
du	21-22	Normal Case, Allowed
uu	22-24	Normal Case, Allowed
uk	24-64	Normal Case, Allowed
k~	64	Normal Case, Allowed

**For variable in(admin):**

Pattern	Line Number	Explanation
~d	29	Normal Case, Allowed
du	29-30	Normal Case, Allowed
uu	-	Normal Case, Allowed
uk	30-64	Normal Case, Allowed
k~	64	Normal Case, Allowed

**For variable in(customer):**

Pattern	Line Number	Explanation
~d	36	Normal Case, Allowed
du	36-37	Normal Case, Allowed
uu	-	Normal Case, Allowed
uk	37-64	Normal Case, Allowed
k~	64	Normal Case, Allowed

**For variable in(high):**

Pattern	Line Number	Explanation
~d	43	Normal Case, Allowed
du	43-44	Normal Case, Allowed
uu	-	Normal Case, Allowed
uk	44-64	Normal Case, Allowed
k~	64	Normal Case, Allowed



## **7.SELENIUM TESTING TOOL**

Selenium is a popular open-source testing tool used for web application testing.

It enables testers to write automated tests in various programming languages to test the functionality of web applications.

Selenium tests can be run on many different browsers and operating systems. Selenium Testing Tool is an excellent tool for automating webapplication testing.

It can be used to test the functionality of web applications on various browsers and operating systems. Selenium Software Testing is a great way to automate your web application testing.

Selenium Record and Playback tool for ease of getting acquainted with Selenium WebDriver. The new Selenium IDE is designed to record your interactions with websites to help you generate and maintain site automation, tests, and remove the need to manually step through repetitive tasks.

## Features include:

- Recording and playing back tests on Firefox and Chrome.
- Organizing tests into suites for easy management.
- Saving and loading scripts, for later playback.
- Support for Selenium 3.

This is the Selenium IDE used to the SHOPPERSSTOP website.



Pros	Cons
Simpler installation than Selenium RC	Installation is more complicated than Selenium IDE
Communicates directly to the browser	Requires programming knowledge
Browser interaction is more realistic	Cannot readily support new browsers
No need for a separate component such as the RC Server	Has no built-in mechanism for logging runtime messages and generating test results
Faster execution time than IDE and RC	

# How to install selenium:

## Selenium ide:

1- Open google chrome

2- Type extensions

3- Search selenium inside

4- Add extension selenium IDE

5- Next use selenium by dragging link of a website which is needed to be tested.

chrome web store

Search extensions and themes

Discover Extensions Themes

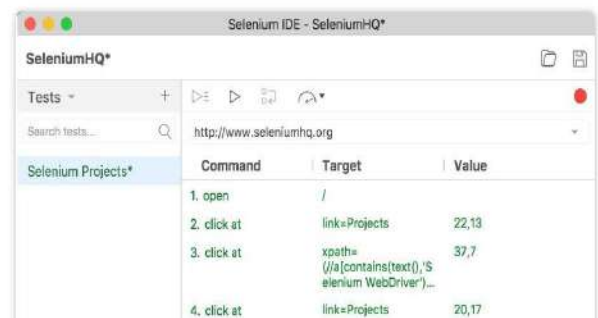
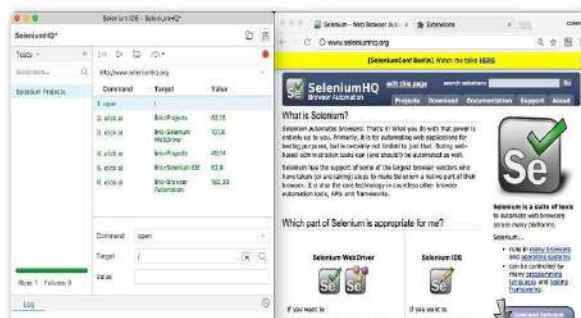


Selenium IDE

Remove from Chrome

seleniumhq.org Featured 3.4 ★ (261 ratings)

Extension Developer Tools 800,000 users



# 1- LOGIN MODULE:

Selenium IDE - login 7

Project: login 7

Tests +

Search tests...

https://www.shoppersstop.com/

Command	Target	Value
7 type	id=mobileNumber	4523472934
8 send keys	id=mobileNumber	\$(KEY_ENTER)
9 type	id=mobileNumber	765767789
10 send keys	id=mobileNumber	\$(KEY_ENTER)
11 click	id=otp-field-1	

Command: type

Target: id=mobileNumber

Value: 765767789

Description:

Log Reference

- 1. open on https://www.shoppersstop.com/ OK 11:52:38
- 2. setWindowSize on 1552x940 OK 11:52:38
- 3. mouseOver on linkText=HOMESTOP OK 11:52:38
- 4. mouseOut on linkText=HOMESTOP OK 11:53:29
- 5. click on css=login OK 11:53:29
- 6. Trying to find css=loginWithMobileForm > row... OK 11:53:30
- 7. type on id=mobileNumber with value 4028472834 OK 11:53:31
- 8. sendKeys on id=mobileNumber with value \$(KEY\_ENTER) OK 11:53:32
- 9. type on id=mobileNumber with value 9843984594 OK 11:53:34

Selenium IDE - login 7\*

Project: login 7\*

Tests +

Search tests...

https://www.shoppersstop.com/

Command	Target	Value
8 ✓ send keys	id=mobileNumber	\$(KEY_ENTER)
9 ✓ type	id=mobileNumber	9843984594
10 ✓ send keys	id=mobileNumber	\$(KEY_ENTER)
11 ✓ click	id=otp-field-1	
12 ✓ type	id=otp-field-1	7
13 ✓ type	id=otp-field-2	4
14 ✓ type	id=otp-field-6	
15 ✓ type	id=otp-field-5	

Command:

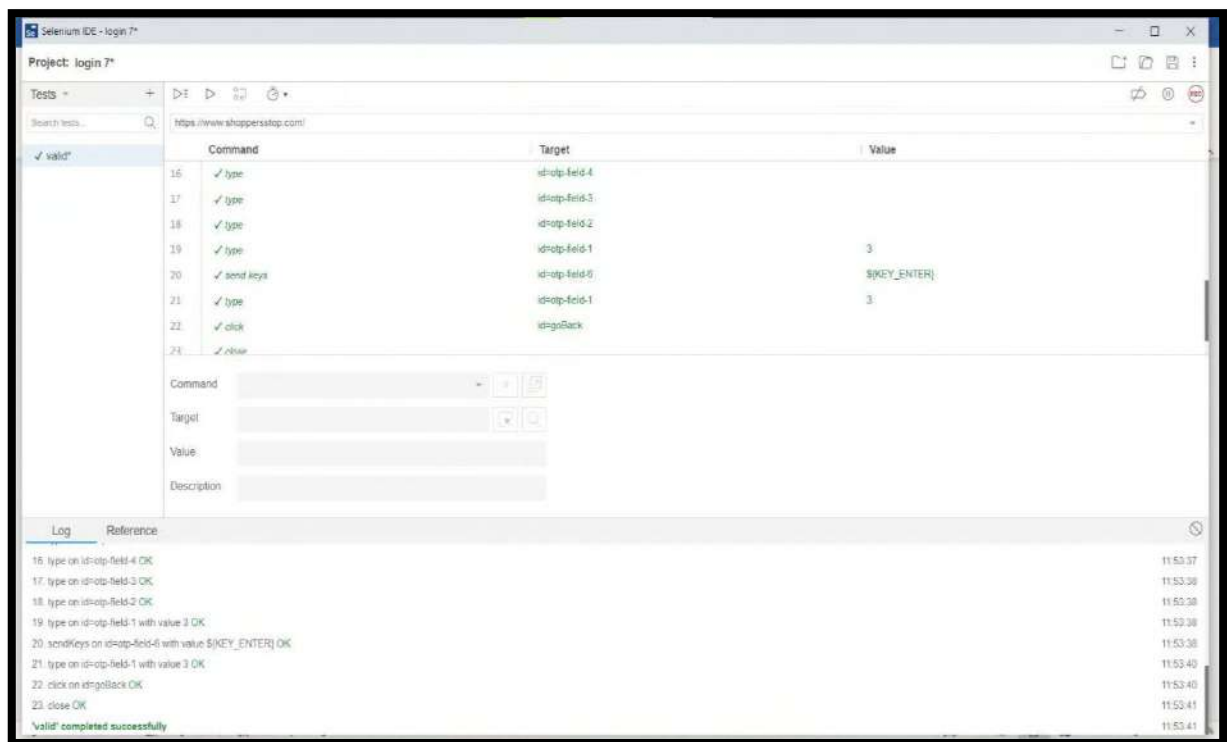
Target:

Value:

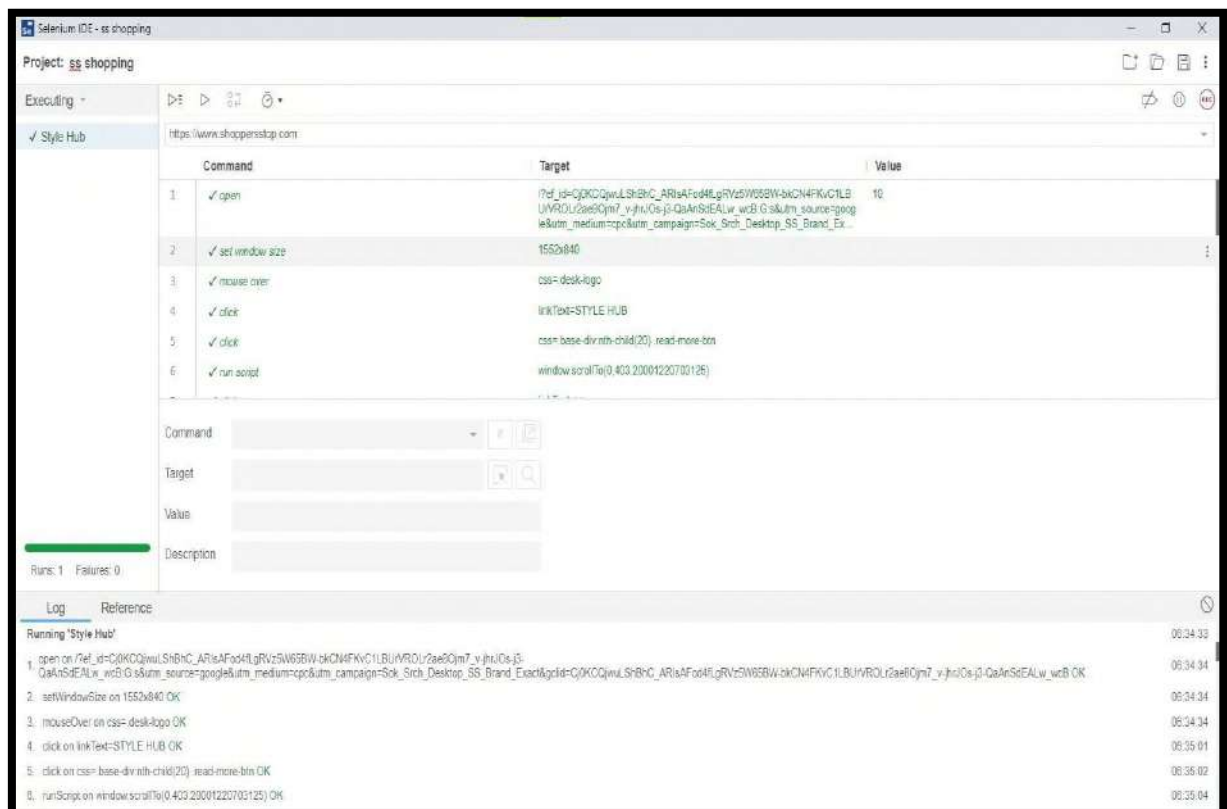
Description:

Log Reference

- 8. sendKeys on id=mobileNumber with value \$(KEY\_ENTER) OK 11:53:32
- 9. type on id=mobileNumber with value 9843984594 OK 11:53:34
- 10. sendKeys on id=mobileNumber with value \$(KEY\_ENTER) OK 11:53:34
- 11. click on id=otp-field-1 OK 11:53:35
- 12. type on id=otp-field-1 with value 7 OK 11:53:36
- 13. type on id=otp-field-2 with value 4 OK 11:53:37
- 14. type on id=otp-field-6 OK 11:53:37
- 15. type on id=otp-field-5 OK 11:53:37
- 16. type on id=otp-field-4 OK 11:53:37



## i. Style Hub Module:



Selenium IDE - ss.shopping

Project: **ss.shopping**

Executing:

✓ Style Hub

https://www.shoppersstop.com

Command	Target	Value
✓ click	linkText=<<	
✓ click	linkText=<<	
✓ click	linkText=<<	
✓ run script	window.scrollTo(0,0)	
✓ click	linkText=<<	
✓ click	css=#menu-navigation-2 > #menu-item-40599 > a	
✓ click	css=#bs-flexisel-nav-right	

Command:

Target:

Value:

Description:

Runs: 1 Failures: 0

Log Reference

Log	Reference
14. click on linkText=<< OK	06:35:16
15. click on linkText=<< OK	06:35:20
16. click on linkText=<< OK	06:35:23
17. click on linkText=<< OK	06:35:26
18. click on css=#menu-navigation-2 > #menu-item-40599 > a OK	06:35:27
19. click on css=#bs-flexisel-nav-right OK	06:35:29
20. click on css=#main-div-rth-child(3) img OK	06:35:32
21. click on css=#wp-image-17229 OK	06:35:32

Selenium IDE - ss.shopping

Project: **ss.shopping**

Executing:

✓ Style Hub

https://www.shoppersstop.com

Command	Target	Value
✓ click	css=#bs-flexisel-nav-right	
✓ click	css=#main-div-rth-child(3) img	
✓ click	css=#wp-image-17229	
✓ select window	handle={\$win7278}	
✓ mouse over	css=#slick-slide00 .dtweb	
✓ run script	window.scrollTo(0,260.79998776296875)	
✓ mouse out	css=#slick-slide00 .dtweb	

Command:

Target:

Value:

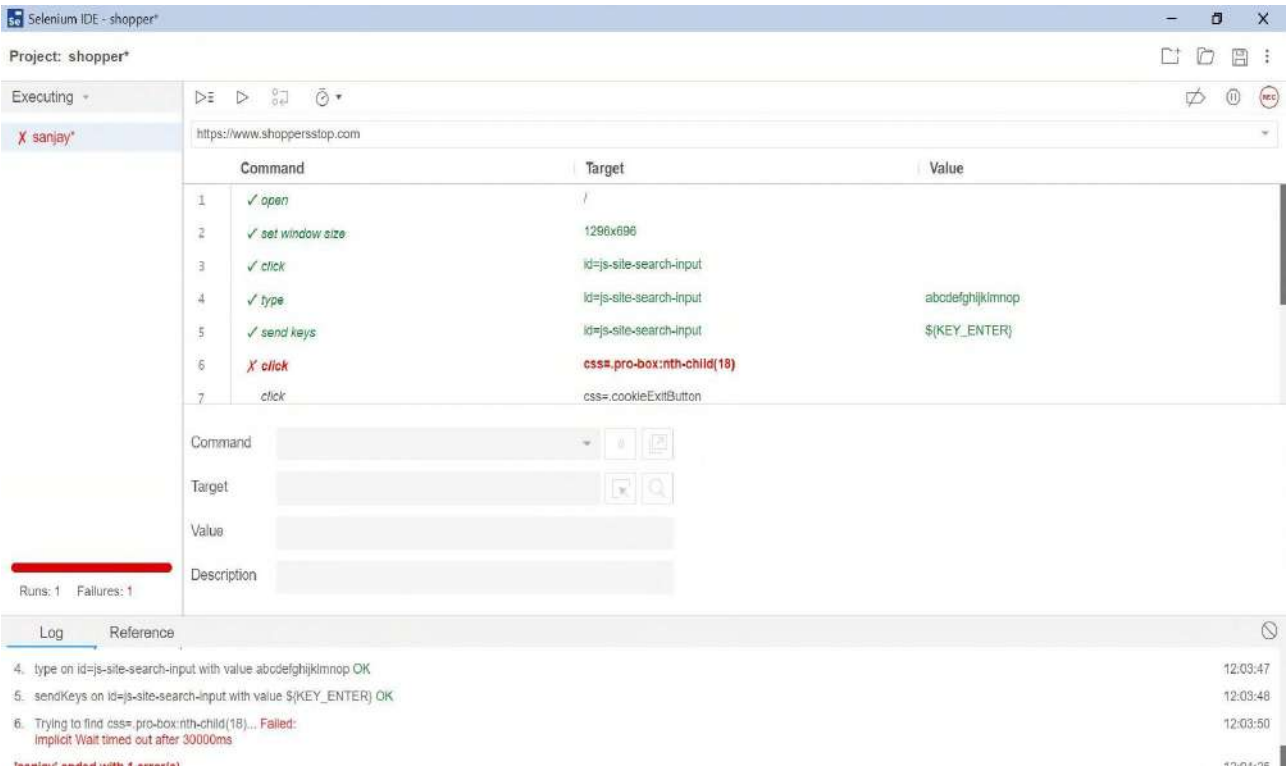
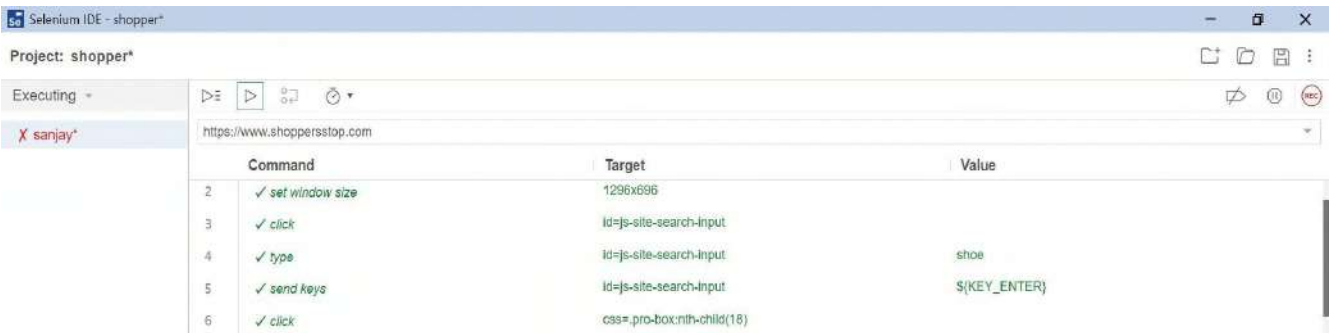
Description:

Runs: 1 Failures: 0

Log Reference

Log	Reference
20. click on css=#main-div-rth-child(3) img OK	06:35:32
21. click on css=#wp-image-17229 OK	06:35:32
22. selectWindow on handle={\$win7278} OK	06:35:35
23. Trying to find css=#slick-slide00 .dtweb... OK	06:35:35
24. runScript on window.scrollTo(0,260.79998776296875) OK	06:36:27
25. runScript on window.scrollTo(0,0) OK	06:36:28
26. mouseOut on css=#slick-slide00 .dtweb OK	06:36:28
'Style Hub' completed successfully	06:36:28

# Even error raised if change in the cases of texts:



## 8. Website Quality Checking:

Tool used to test quality of a website is sort site it checks entire website.

**SortSite** is a web crawler that scans entire websites for quality issues including accessibility, browser compatibility, broken links, legal compliance, search optimization, usability and web standards compliance.

Test standards and checkpoints

Quality tests run on each page include:

- Accessibility - W3 WCAG 1.0, 2.0 and Section 508 standards
- Browser compatibility - check cross-browser compatibility of HTML, CSS and JavaScript (i.e. find code that doesn't work in all browsers)
- Broken links - checks for broken links, missing images and HTTP protocol violations
- Search engine guidelines - Yahoo, Microsoft and Google guidelines - websites violating the guidelines may be removed from the Google index
- Usability - Usability.gov peer-reviewed web usability guidelines[1]
- Web standards - validation of HTML, XHTML and CSS



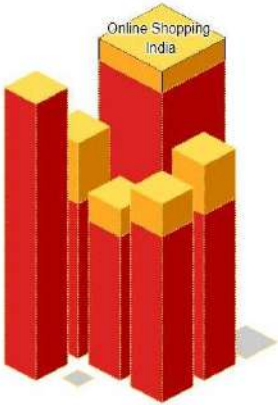
# Summary Report:

## SortSite Scan Report

Scan completed for first 10 pages of  
[https://www.shoppersstop.com/?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=SHOP\\_13245\\_adyogi\\_Brand\\_Search\\_Engage\\_Desktop\\_Exact-20307001336&gad\\_source=1&gclid=CjwKCAjw\\_e2wBhAEEiwAyFFFo-WNzkHdtBwv5auBsNAZjCIWWE-liC7hhjEhM-ixZepfxRmnsfpfTBoCDmsQAvD\\_BwE](https://www.shoppersstop.com/?utm_source=google&utm_medium=cpc&utm_campaign=SHOP_13245_adyogi_Brand_Search_Engage_Desktop_Exact-20307001336&gad_source=1&gclid=CjwKCAjw_e2wBhAEEiwAyFFFo-WNzkHdtBwv5auBsNAZjCIWWE-liC7hhjEhM-ixZepfxRmnsfpfTBoCDmsQAvD_BwE)



This tab shows where errors occur on a map of the site. Each page is represented by a tower, with the widest towers one click from the home page and towers getting narrower the more clicks a page is from the home page. The tallest towers indicate pages with most errors. Hover over the tops of the towers to see page details.



### Guideline

#### Accessibility

- WAI-ARIA 1.2
- JAWS Screen Reader Compatibility
- NVDA Screen Reader Compatibility
- VoiceOver Screen Reader Compatibility
- Scanner Blocked
- PDF/UA - Matterhorn Protocol 1.02
- Section 508 of the Rehabilitation Act - 82 FR 5790 (2017)
- Web Content Accessibility Guidelines 2.0 (WCAG 2.0 - 2008)
- Web Content Accessibility Guidelines 2.1 (WCAG 2.1 - 2018)
- Web Content Accessibility Guidelines 2.2 (WCAG 2.2 - 2023)

### Status

- ☑ Checked
- Not Checked
- Not Checked
- Not Checked
- ☑ Checked
- Not Checked
- ☑ Checked
- Not Checked
- Not Checked
- ☑ Checked

### Browser Compatibility

Android Browser Compatibility	✓ Checked latest version
Chrome Browser Compatibility	✓ Checked latest version
Edge Browser Compatibility	✓ Checked latest version
Internet Explorer Browser Compatibility	Not Checked
Firefox Browser Compatibility	✓ Checked latest version
Opera Browser Compatibility	✓ Checked latest version
Safari Browser Compatibility	✓ Checked latest version
iPhone/iPad Browser Compatibility	✓ Checked from version 15

### Errors

Server Configuration	✓ Checked
Content Issues	✓ Checked - No Spell Check
Blocked Links	✓ Checked
User Defined Errors	✓ Checked
HTTP Status Codes	✓ Checked
IETF RFCs	✓ Checked
Script Errors	✓ Checked

### Search Engine Guidelines

Google Webmaster Guidelines	✓ Checked
Google SEO Starter Guide	✓ Checked
Google Blogs and Webmaster Videos	✓ Checked
Bing Webmaster Guidelines	✓ Checked

### Usability

Readability Guidelines	✓ Checked
Usability.gov: Research-Based Web Design & Usability Guidelines	✓ Checked
W3C Usability Best Practices	✓ Checked

### Web Standards

W3C CSS Validation	✓ Checked
W3C Deprecated Features	✓ Checked
W3C HTML Validation	✓ Checked
HTML Living Standard	✓ Checked

# SUMMARY REPORT:

Home > Products > SortSite > Results

Dashboard

Site map

What was tested

ad\_source=1&gclid=CiwKCAiw\_e2w8hAEEiwAyFFFo-WNzkHdtBww5auBsNAZiCIWWE-liC7hjEhM-TBoCDmsOAvD\_BwE

Category	Issues	Pages	Benchmark
Overall Quality	<div><div></div></div>	6 pages with quality issues	❌ 14% have issues, worse than average
<a href="#">Errors</a>	<div><div></div></div>	6 pages with broken links or other errors	✅ 14% have issues, better than average
<a href="#">Accessibility</a>	<div><div></div></div>	6 pages with accessibility problems	✅ 14% have issues, better than average
<a href="#">Compatibility</a>	<div><div></div></div>	0 pages with browser specific issues	✅ 0% have issues, better than average
<a href="#">Search</a>	<div><div></div></div>	4 pages with search engine issues	✅ 10% have issues, better than average
<a href="#">Standards</a>	<div><div></div></div>	6 pages have W3C standards issues	✅ 14% have issues, better than average
<a href="#">Usability</a>	<div><div></div></div>	6 pages with usability issues	✅ 14% have issues, better than average
Totals		44 pages and files checked	

The trial version is limited to checking 10 pages and images.

## Errors occurred:

Home > Products > SortSite > Results

❌ Broken links - Some pages contain links that don't work.

✅ Server configuration - No issues found.

✅ ASP, ASP.NET and PHP script errors - No issues found.

✅ Internet RFCs - No issues found.

Priority	Description and URL	Guideline and Line#	Count
Priority 1			
3 issues on 6 pages			
❌	This link is broken. The name of the target web server not found in DNS.	Broken Link	1 pages
❌	This link is broken. The page could not be found on the target web server.	<a href="#">Broken Link HTTP 404</a>	1 pages
❌	This link is broken. The src or href is an empty string.	Broken Link	6 pages
Informative			
These messages are for information only and do not indicate errors or conformance problems			
ⓘ	Spell checking was not enabled for this scan.		1 pages
⌵	Expand all 4 issues		

# Accessibility:

[https://www.shopensip.com/?utm\\_source=google&utm\\_medium=google&utm\\_campaign=SHOP\\_13245\\_ad&utm\\_brand=Search+Engage+Desktop\\_Fixed&utm\\_term=13245&utm\\_source=16y3ld=QwIC4w\\_e2N0h4FEw2vFTFo7NkH4Bw5auBd47CUMWF4C7h4E4k\\_uZed4Bren4q71e0DmeG4vL\\_Bu4](https://www.shopensip.com/?utm_source=google&utm_medium=google&utm_campaign=SHOP_13245_ad&utm_brand=Search+Engage+Desktop_Fixed&utm_term=13245&utm_source=16y3ld=QwIC4w_e2N0h4FEw2vFTFo7NkH4Bw5auBd47CUMWF4C7h4E4k_uZed4Bren4q71e0DmeG4vL_Bu4)

Errors Accessibility Compatibility Search Standards Usability

This tab shows accessibility issues, indicating problems for older users, people with disabilities or accessibility needs. Automated testing cannot detect all accessibility issues, so should be used alongside human testing.

Level	WCAG 2.2	Section 508 - 2017	Key
A			Pages with level A issues are unusable for some people
AA			Pages with level AA issues are very difficult to use
AAA			Pages with level AAA issues can be difficult to use

Priority	Description and URL	Guideline and Line#	Count
----------	---------------------	---------------------	-------

## Level A

10 issues on 6 pages

		<a href="#">alt</a> text should not be an image file name.	<a href="#">WCAG 2.2 A.F30 Section 508 (2017) A.F30</a>	3 pages
		HTML form control has no accessible name.	<a href="#">WCAG 2.2 A.F30 Section 508 (2017) A.F30</a>	4 pages
		Identify row and column headers in data tables using <a href="#">th</a> elements, and mark layout tables with <a href="#">role=presentation</a> .	<a href="#">WCAG 2.2 A.F91 Section 508 (2017) A.F91</a>	1 pages
		<a href="#">iframe</a> , and <a href="#">frame</a> elements must have a <a href="#">title</a> attribute.	<a href="#">WCAG 2.2 A.4.1.2 Section 508 (2017) A.4.1.2</a>	1 pages
		<a href="#">img</a> elements must have an accessible name.	<a href="#">WCAG 2.2 A.F95 Section 508 (2017) A.F95</a>	4 pages
		Links must have an accessible name.	<a href="#">WCAG 2.2 A.F99 Section 508 (2017) A.F99</a>	6 pages
		The element <a href="#">button</a> must not appear as a descendant of the <a href="#">a</a> element.	<a href="#">HTML5 WCAG 2.2 A.4.1.2 Section 508 (2017) A.4.1.2</a>	6 pages
		The interactive element <a href="#">button</a> must not appear as a descendant of the <a href="#">a</a> element.	<a href="#">HTML5 WCAG 2.2 A.4.1.2 Section 508 (2017) A.4.1.2</a>	6 pages
		This element uses JavaScript to behave like a link. Links like this cannot be tabbed to from the keyboard and are not read out when screen readers list the links on a page.	<a href="#">WCAG 2.2 A.F42 Section 508 (2017) A.F42</a>	6 pages
		This <a href="#">input</a> button has no <a href="#">value</a> attribute and no label.	<a href="#">WCAG 2.2 A.4.1.2 Section 508 (2017) A.4.1.2</a>	6 pages

[Home](#) » [Products](#) » [SortSite](#) » [Results](#)

		Identify row and column headers in data tables using <a href="#">th</a> elements, and mark layout tables with <a href="#">role=presentation</a> .	<a href="#">WCAG 2.2 A.F91 Section 508 (2017) A.F91</a>	1 pages
		<a href="#">iframe</a> , and <a href="#">frame</a> elements must have a <a href="#">title</a> attribute.	<a href="#">WCAG 2.2 A.4.1.2 Section 508 (2017) A.4.1.2</a>	1 pages
		<a href="#">img</a> elements must have an accessible name.	<a href="#">WCAG 2.2 A.F95 Section 508 (2017) A.F95</a>	4 pages
		Links must have an accessible name.	<a href="#">WCAG 2.2 A.F99 Section 508 (2017) A.F99</a>	6 pages
		The element <a href="#">button</a> must not appear as a descendant of the <a href="#">a</a> element.	<a href="#">HTML5 WCAG 2.2 A.4.1.2 Section 508 (2017) A.4.1.2</a>	6 pages
		The interactive element <a href="#">button</a> must not appear as a descendant of the <a href="#">a</a> element.	<a href="#">HTML5 WCAG 2.2 A.4.1.2 Section 508 (2017) A.4.1.2</a>	6 pages
		This element uses JavaScript to behave like a link. Links like this cannot be tabbed to from the keyboard and are not read out when screen readers list the links on a page.	<a href="#">WCAG 2.2 A.F42 Section 508 (2017) A.F42</a>	6 pages
		This <a href="#">input</a> button has no <a href="#">value</a> attribute and no label.	<a href="#">WCAG 2.2 A.4.1.2 Section 508 (2017) A.4.1.2</a>	6 pages

## Level AA

5 issues on 6 pages

		Cannot use <a href="#">aria-label</a> or <a href="#">aria-labelledby</a> on elements and roles that prohibit naming.	<a href="#">HTML5 ARIA 1.2</a>	6 pages
		Do not use the <a href="#">meta viewport</a> tag to disable zoom.	<a href="#">WCAG 2.2 AA 1.4.4 Section 508 (2017) AA 1.4.4</a>	6 pages
		Ensure that text and background colors have enough contrast.	<a href="#">WCAG 2.2 AA 1.4.3 Section 508 (2017) AA 1.4.3</a>	4 pages
		For <a href="#">input type=password</a> elements, set the <a href="#">autocomplete</a> attribute to <a href="#">new-password</a> or <a href="#">current-password</a> in order to identify input purpose.	<a href="#">WCAG 2.2 1.3.5</a>	6 pages
		The CSS outline or border style on this element makes it difficult or impossible to see the link focus outline.	<a href="#">WCAG 2.2 AA F78 Section 508 (2017) AA F78</a>	6 pages

## Level AAA

3 issues on 4 pages

		Avoid specifying a new window as the target of a link with <a href="#">target=_blank</a> .	<a href="#">WCAG 2.2 AAA F22</a>	3 pages
		Clickable targets must be at least 44 by 44 CSS pixels.	<a href="#">WCAG 2.2 2.5.5</a>	3 pages
		Ensure that text and background colors have a 7:1 contrast ratio.	<a href="#">WCAG 2.2 AAA 1.4.6</a>	1 pages
		Expand all 18 issues		

## Compatibility:

Home > Products > SortSite > Results

Summary Issues Pages

<https://rimsight.sagepub.com/doi/full/10.1177/1043986210384510>

Errors Accessibility Compatibility Search Standards Usability

This tab shows pages that exhibit browser-specific behavior, or trigger browser bugs.

Browser	Edge	Firefox	Safari	Opera	Chrome	iOS	Android	
Version	120	120	17	105	120	≤ 16	17	120
Critical Issues								Missing content or functionality
Major Issues								Major layout or performance problems
Minor Issues								Minor layout or performance problems

Priority Description and URL Guideline and Line# Count

Expand all 0 issues

## Search

Home > Products > SortSite > Results

Summary Issues Pages

[https://www.shoppersstop.com/?utm\\_source=google&utm\\_medium=ppc&utm\\_campaign=SHOP\\_13245\\_3d/99\\_Brand\\_Search\\_Engine\\_Dept&utm\\_product=3670013358433\\_source=1&gclid=CjwKAE42N6BvEENwUwFF6VjN0dM0BwSaubtLUQ3WVIEUC7HjEhM.02e6d8Rm9qdUThCf0m0A0U\\_BwE](https://www.shoppersstop.com/?utm_source=google&utm_medium=ppc&utm_campaign=SHOP_13245_3d/99_Brand_Search_Engine_Dept&utm_product=3670013358433_source=1&gclid=CjwKAE42N6BvEENwUwFF6VjN0dM0BwSaubtLUQ3WVIEUC7HjEhM.02e6d8Rm9qdUThCf0m0A0U_BwE)

Errors Accessibility Compatibility Search Standards Usability

This tab shows search engine guideline violations, and pages that don't follow search optimization best practices.

This tab is available in SortSite Professional, SortSite Developer and OnDemand, but not in SortSite Standard edition.

Priority	Description and URL	Guideline and Link	Count
<b>Priority 1</b> 2 issues on 4 pages			
❌	Google Search Guidelines - Some pages violate these guidelines.		
❌	Bing Search Guidelines - Some pages violate these guidelines.		
❌	Yahoo Search Guidelines - Some pages violate these guidelines.		
✅	Robots.txt Guidelines - No issues found.		
✅	Search Best Practices - No issues found.		
<b>Priority 1</b> 2 issues on 4 pages			
❌	Google and Bing are unable to follow JavaScript links. Every page must be reachable by a static text link.	<a href="#">Google</a> <a href="#">Bing</a>	3 pages
❌	This page has more than one h1 element, which violates Bing webmaster guidelines.	<a href="#">Bing</a>	1 page
<b>Priority 3</b> 1 issues on 1 pages			
⚠️	Search engines may penalize invisible text where text color is nearly identical to the background color.	<a href="#">Google</a> <a href="#">Yahoo</a> <a href="#">Bing</a>	1 pages
<b>Informative</b> These messages are for information only and do not indicate errors or conformance problems			
ℹ️	Search engines cannot index areas of sites that require a log in.	<a href="#">Google</a> <a href="#">Bing</a>	5 pages
ℹ️	Search engines will not index this page because it has a noindex meta tag.	<a href="#">Robots.txt</a>	1 pages
Expand all 5 issues			



# Standards:

Home > Products > SortSite > Results

W3C HTML/XHTML Validation - Some pages fail validation.

W3C CSS Validation - Some pages fail validation.

W3C Dispreccated Featunes - No Issues found.

Priority	Description and URL	Guideline and Line#	Count
Priority 1			
64 Issues on 8 pages			
	An <code>img</code> element must have an <code>alt</code> attribute, except under certain conditions. For details, consult guidance on providing text alternatives for images.	<a href="#">HTML5</a>	5 pages
	Attribute <code>accept</code> not allowed on element.	<a href="#">HTML5</a>	6 pages
	Attribute <code>commandname</code> not allowed on element.	<a href="#">HTML5</a>	6 pages
	Attribute <code>controlslist</code> not allowed on element.	<a href="#">HTML5</a>	6 pages
	Attribute <code>101010110_</code> not allowed on element.	<a href="#">HTML5</a>	3 pages
	Attribute <code>secure-login</code> not allowed on element.	<a href="#">HTML5</a>	6 pages
	Bad start tag <code>img</code> in <code>hscrypt</code> in <code>head</code> .	<a href="#">HTML5</a>	3 pages
	Bad value for attribute <code>id</code> .	<a href="#">HTML5</a>	6 pages
	Bad value for attribute <code>src</code> .	<a href="#">HTML5</a>	4 pages
	Bad value for attribute <code>width</code> .	<a href="#">HTML5</a>	6 pages
	Consider avoiding viewport values that prevent users from resizing documents.	<a href="#">HTML5</a>	6 pages
	CSS Validation Error.	CSS Validation	1 pages
	Duplicate attribute <code>id</code> .	<a href="#">HTML5</a>	3 pages
	Duplicate <code>id</code> - the same ID is used on more than one element.	<a href="#">HTML5</a>	6 pages
	Element <code>div</code> not allowed as child element in this context.	<a href="#">HTML5</a>	6 pages
	Element <code>globalmessages</code> not allowed as child element in this context.	<a href="#">HTML5</a>	2 pages
	Element <code>h4</code> not allowed as child element in this context.	<a href="#">HTML5</a>	1 pages
	Element <code>img</code> is missing one or more required attributes.	<a href="#">HTML5</a>	3 pages
	Element <code>input</code> not allowed as child element in this context.	<a href="#">HTML5</a>	6 pages
	Element <code>meta</code> is missing one or more required attributes.	<a href="#">HTML5</a>	3 pages
	Element <code>option</code> without attribute <code>label</code> must not be empty.	<a href="#">HTML5</a>	4 pages

# Usability:

Home > Products > SortSite > Results

Accessibility > Usability

Usability.gov Guidelines - Some pages violate these guidelines.

W3C Best Practices - Some pages violate these guidelines.

Readability - No issues found.

Priority	Description and URL	Guideline and Line#	Count
Priority 2			
4 Issues on 6 pages			
	Keep URLs shorter than 78 characters so they don't wrap when emailed.	<a href="#">W3C</a>	1 pages
	Minimize the time required to download a Web site's pages.	<a href="#">Usability.gov 2.9</a>	2 pages
	Omitting <code>img width</code> or <code>height</code> attributes makes the page layout jump about as images load.	<a href="#">Usability.gov 14.3 W3C</a>	6 pages
	On long pages, provide a list of contents with links that take users to the corresponding content farther down the page.	<a href="#">Usability.gov 7.3</a>	2 pages
Priority 3			
3 Issues on 6 pages			
	Radio buttons are easier to use than drop-downs when there are 5 choices or fewer.	<a href="#">Usability.gov 13.9</a>	6 pages
	Use at least a 12-point font on all web pages.	<a href="#">Usability.gov 11.8</a>	1 pages
	Use option groups when a drop down list has more than 10 items.	<a href="#">Usability.gov 13.12</a>	4 pages
Priority 4			
1 Issues on 6 pages			
	Make sure one radio button in a radio button group is always selected. If users can choose not to activate any of the radio button choices, provide a choice labeled 'None'.	<a href="#">Usability.gov 13.23</a>	6 pages
	Expand all 8 issues		

# Pages:

[Home](#) > [Products](#) > [SortSite](#) > [Results](#)

Type	URL	Description
<b>Documents</b>		
» <b>HTML</b>	6 pages	HTML pages
» <b>PDF</b>	0 files	Adobe Acrobat PDFs
» <b>Other</b>	0 files	Other documents
<b>Images</b>		
» <b>GIF</b>	0 images	GIF images
» <b>PNG</b>	0 images	PNG images
» <b>JPEG</b>	0 images	JPEG images
» <b>SVG</b>	0 images	SVG images
» <b>Other</b>	0 images	Other image types
<b>Misc.</b>		
» <b>CSS</b>	2 files	Cascading Style Sheets
» <b>JavaScript</b>	0 files	JavaScript files
» <b>Fonts</b>	0 files	Web fonts
» <b>Media</b>	0 files	Audio and video files
» <b>Data</b>	0 files	Feeds and data
» <b>Other</b>	0 files	Other file types
<b>Links</b>		
» <b>Special</b>	1 files	Special files like robots.txt

9.

Test Log Report:

Test Log Identifier TL01

Description:

Function 3.6 in SRS v1.1. The function is tested in Offline environment with Windows System.

Activity and Event Entries

1. Date: 14/04/2024

2. Author of test: SANJAY

3. Test case: For Login module.

4. Name of the personnel involved in testing:  
SANJAY, PRASANTH, SIDHART JAIN.

5. For each execution, record the results and mention pass/fail status

6. Username and password.

#For each possibility the status of test case is displayed in the table.

#Report an anomalous unexpected event before and after execution.



The function was tested with the following inputs:

Testcase ID	Input		Expected Output	Status
	User_ID	Password		
TC01	Sanjay_26	-	Prompt message tofill the password blank	Fail
TC02	ABC	123	Error message	Fail
TC03	SanjayS_2	Abc.123	Successful	Pass
TC04	ABC	A123...	Error message	Fail
TC05	Sanjay_1	Abc@12	Successful	Pass

10.

## TEST INCIDENT REPORT

Test Incident Report Identifier: TL01

Summary:

Function 3.6 in SRS v1.1. Login Module and Test Log  
TL01 Incident

Description:

It describes the following

i.Date and Time: 14/04/2024, 3:00PM

ii. Testing personal name: Sanjay , prasanth , sidhart jain

iii.Environment: Online Testing tool

iv. Testing Input

v. Expected Output

vi. Actual Output

vii. Anomalies detected during testing

viii. Attempt to execute same test

Testcase ID	Input		Expected Output	Actual Output	Detect anomaly	Attempt Repeat
	User_ID	Password				
TC01	sanjay_26	-	Prompt message to fill the password blank	Prompt message to fill the password blank	It should give error to fill the blank	3
TC02	ABC	123	Error message	Error message	It should give error to enter the valid credentials.	2
TC03	Prasanth_2	Abc.123	Successful	Successful	Nil	1
TC04	@123	A123...	Error message	Error message	It should give error to enter the valid credentials.	3
TC05	sidhart_1	Abc@12	Successful	Successful	Nil	1

## **CONCLUSION:**

In conclusion, testing the Shoppers Stop website is a crucial aspect of software testing to ensure that it functions correctly and provides a seamless shopping experience for its users. By testing the website's functionality, usability, performance, security, compatibility, and accessibility, software testers can identify any defects, bugs, or issues before the website is released to the public. This can help prevent negative feedback, customer dissatisfaction, and lost revenue. Furthermore, testing helps improve the overall quality of the website and enhances the user experience, which is essential for the success of the business. Ultimately, testing the Shoppers Stop website ensures that users can shop with confidence and trust in the website's reliability and security.

## **REFERENCES:**

**Cem Kaner 1988 foundations of software testing.**

**Corey sandler 2011 testing computer software.**

**Paul c. jorgenson 2013 software testing a craft.**

**James A whittaker 2012 how google test software.**