VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI

A Mini Project Report
On

"Automation Testing of an E-Commerce Website - WROGN "

Submitted in partial fulfilment of the requirements for the award of 6th Sem of

Bachelor of Engineering in Information Science and Engineering

Submitted by

KUNAL KUSHALAPPA P N[1VE21IS025]

MOHAMMED PARVEEZ [1VE21IS034]

SRI PRANAV SPOORTHY[1VE21IS053]

Under the guidance of

Mr. CHETHAN C
Asst. Professor
Dept. of ISE

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

SRI VENKATESHWARA COLLEGE OF ENGINEERING

Accredited by NAAC & NBA*
SVCE BENGALURU
SRI VENKATESHWARA COLLEGE OF ENGINEERING
— Affiliated to VTU, Approved by AICTE, Recognised by UGC u/s 2(f) & 12(B)—

Affiliated to VTU Belgaum & Approved by AICTE New Delhi) An ISO 9001: 2008 Certified
Kempegowda International Airport road, Vidyanagar
Bengaluru, Karnataka, India-562157

2023 – 2024

SRI VENKATESHWARA COLLEGE OF ENGINEERING

(Affiliated by A.I.C.T.E and approved by V.T.U)

Department of Information Science and Engineering

Accredited by NAAC & NBA*

SVCE BENGALURU

SRI VENKATESHWARA COLLEGE OF ENGINEERING

— Affiliated to VTU, Approved by AICTE, Recognised by UGC u/s 2(f) & 12(B)—

CERTIFICATE

This is to certify that Mini Project entitled **"Automation Testing of an E-Commerce Website - WROGN"** is submitted by **KUNAL KUSHALAPPA PN [1VE21IS025], MOHAMMED PARVEEZ[1VE21IS034] , SRI PRANAV SPOORTY [1VE21IS053]** in partial fulfilment of the requirement for VI semester, Bachelor of Engineering in Information Science and Engineering, Visvesvaraya Technological University, Belgaum for the academic year **2023-2024**

….....	……	……..
Signature of the Guide	**Signature of the HOD**	**Signature of the Principal**
Mr. Chethan C	**Dr. Sunil Kumar K N**	**Dr. Nageswara Guptha M**
Asst. Prof, Dept. of ISE	**Prof, Dept. of ISE**	**SVCE, Bangalore**
SVCE, Bangalore	**SVCE, Bangalore**	

External Viva:

Name of the Examiners: **Signature with Date**

1.

2.

INSTITUTE VISION

To be a premier institute for addressing the challenges in global perspective.

INSTITUTE MISSION

M-1: Nurture students with professional and ethical outlook to identify needs, analyse, design and innovate sustainable solutions through lifelong learning in service of society as individual or a team.

M-2: Establish state-of-the-art Laboratories and Information Resource centre for education and research.

M-3: Collaborate with Industry, Government Organization and Society to align the curriculum and outreach activities.

DEPARTMENT VISION

Excellence in Information Science and Engineering Education, Research and Development

DEPARTMENT MISSION

M-1: Accomplish academic achievement in Information Science and Engineering through student - centered creative teaching learning, qualified faculty and staff members, assessment and effective ICT.

M-2: Establish a Center of Excellence in a various verticals of Information science and engineering to encourage collaborative research and Industry-institute collaboration.

M-3: Transform the engineering students to socially responsible, ethical and technically competent and Value added professional or entrepreneur through holistic education.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO-1: KNOWLEDGE

Information Science and Engineering Graduates will have professional technical career in inter disciplinary domains providing innovative and sustainable solutions using modern tools.

PEO-2: SKILLS

Information Science and Engineering Graduates will have effective communication, leadership, team building, problem solving, decision making and creative skills.

PEO-3: ATTITUDE

Information Science and Engineering Graduates will practice ethical responsibilities towards their peers, employers and society.

TABLE OF CONTENTS

TABLE OF FIGURES

Chapter 1

INTRODUCTION

Software Testing

Software testing is an essential practice in the software development lifecycle, focused on evaluating and verifying that a software application or system functions correctly and meets its specified requirements. The primary objective of software testing is to detect and fix defects or issues before the software is released, thereby ensuring that it operates reliably and performs well under various conditions. This process involves executing the software in a controlled environment to assess its functionality, usability, performance, and security. Testing methods include functional testing, which verifies that the software performs its intended functions; performance testing, which evaluates how the software behaves under stress or heavy load; and security testing, which assesses the application's vulnerability to threats and attacks. Effective software testing requires a well-defined testing strategy, including the creation of detailed test plans and cases that cover various scenarios and user interactions. By identifying defects early in the development process, software testing helps to minimise the risk of software failure, improve user satisfaction, and reduce the overall cost of development and maintenance.

Figure 1.1 Software Testing

Software testing is a crucial aspect of the software development lifecycle, designed to ensure that software applications meet their specified requirements, function correctly, and are free from defects. This process involves a systematic evaluation of software to identify and rectify issues before it is released to end-users, ultimately aiming to deliver a high-quality product.

The primary goals of software testing include identifying defects or bugs, ensuring that the software meets the required quality standards, verifying that it adheres to specified requirements, improving its reliability, and enhancing the overall user experience. Defects can arise from coding errors, integration issues, or discrepancies with requirements, and addressing these issues is vital for producing a stable and functional application.

Testing methodologies can be broadly categorized into manual and automated approaches. Manual testing involves human testers executing test cases without the aid of automated tools. Testers interact with the software in a manner similar to end-users to verify its functionality and identify defects. Manual testing is particularly valuable for exploratory testing, usability assessments, and scenarios where automation may not be feasible or cost-effective. It provides insights into the user experience and can uncover issues that automated scripts might miss.

Automated testing, on the other hand, utilizes software tools and scripts to automatically execute test cases. This approach allows for the efficient and consistent execution of repetitive tasks and regression tests. Automated tests can run quickly and frequently, which is beneficial for continuous integration and delivery practices. Automation is particularly useful for large-scale applications where manual testing would be time-consuming and prone to human error.

The software testing process typically involves several phases, starting with test planning. During this phase, testers develop a strategy for testing, which includes defining the scope, objectives, resources, timelines, and deliverables. Test planning ensures that testing efforts are well-organized and aligned with project goals.

Automation Testing

Automation testing is a technique that utilises specialised tools and scripts to automatically execute test cases, compare actual outcomes with expected results, and generate reports on discrepancies. This approach significantly enhances the efficiency and accuracy of the testing process compared to manual methods. Automated tests can be run quickly and repeatedly, making them particularly effective for repetitive tasks, large-scale testing, and regression tests. By automating these processes, organisations can achieve faster test execution, reduce the risk of human error, and ensure consistent results across different test environments and configurations.

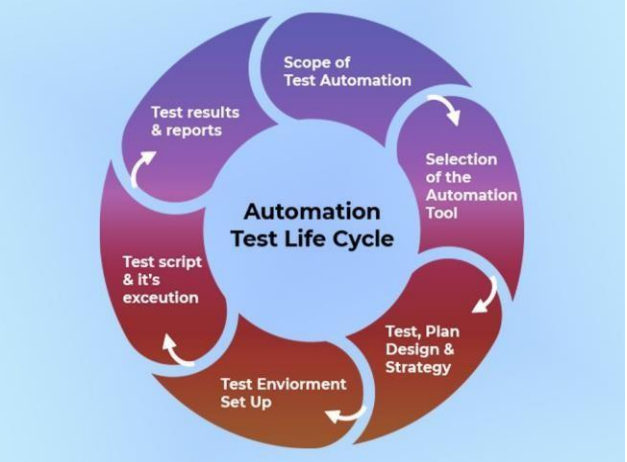

Figure 1.2 Automation Testing Lifecycle

The integration of automation testing into Continuous Integration/Continuous Deployment (CI/CD) pipelines further streamlines the development and deployment process. Automation tools, such as Selenium or QTP, work with test frameworks to provide a structured approach to test creation and management. Although automation testing requires initial setup and ongoing maintenance to keep pace with application changes, its benefits—such as increased test coverage, faster feedback, and higher software quality—make it an essential practice in modern software development. Additionally, automation testing enables rapid validation of new features and bug fixes, allowing teams to deliver robust, high-quality software more efficiently and effectively. It also supports scalability, as automated tests can be easily adapted to accommodate growing applications and evolving business needs, ensuring that software remains reliable and competitive in a fast-paced market.

Web Automation Tool - Selenium

Selenium is a powerful open-source tool designed for automating web applications, primarily for testing purposes, though it can also be used to automate repetitive web tasks. Originally created in 2004 by Jason Huggins, Selenium has evolved significantly and now supports a range of programming languages and browser platforms.

The core component of Selenium is Selenium WebDriver, which enables interaction with web browsers through a programming interface. WebDriver allows for the creation and execution of test cases by simulating user actions and verifying the behavior of web applications. It works by communicating directly with browsers through browser-specific drivers, such as ChromeDriver for Google Chrome and GeckoDriver for Mozilla Firefox. This direct interaction with browsers provides greater control and stability compared to the older Selenium RC (Remote Control) approach.

WebDriver supports several programming languages, including Java, C#, Python, Ruby, and JavaScript, offering an object-oriented API that allows users to perform a wide range of interactions. These interactions include advanced operations like drag-and-drop, multi-touch actions, and handling browser pop-ups. Its flexibility and robust feature set make it suitable for a variety of testing scenarios.

Selenium IDE (Integrated Development Environment) is another component of Selenium. It is a browser extension available for Firefox and Chrome that enables users to record and replay interactions with web applications. This allows users to create simple test scripts without requiring any programming knowledge. Selenium IDE is useful for creating basic tests and prototyping, thanks to its graphical interface that records user actions and generates scripts. However, it is limited in terms of scalability and flexibility when compared to Selenium WebDriver.

The third component, Selenium Grid, provides a solution for distributing tests across multiple machines and browsers, facilitating parallel execution of tests. This is particularly useful for large-scale testing scenarios where running tests concurrently can significantly reduce the overall testing time. Selenium Grid supports multiple versions of browsers and operating systems, making it ideal for cross-browser testing.

Selenium WebDrivers

Selenium WebDriver is a critical component of the Selenium suite, designed to offer a more effective and sophisticated method for automating web browsers. It enables users to programmatically control a web browser, making it a vital tool for testing and automating interactions with web applications. Unlike its predecessor, Selenium RC (Remote Control), WebDriver was built to overcome several limitations and offer a more direct and reliable approach.

At its core, Selenium WebDriver follows a client-server architecture. The client libraries are available in various programming languages, including Java, C#, Python, Ruby, and JavaScript. These libraries provide the API needed to write scripts that interact with web browsers. The client libraries serve as the entry point for creating and executing tests, offering a straightforward way to define interactions with web elements and browser behaviors.

WebDriver communicates with browsers through specific browser drivers. Each major browser has its own driver, such as ChromeDriver for Google Chrome, GeckoDriver for Mozilla Firefox, SafariDriver for Safari, and EdgeDriver for Microsoft Edge. These browser drivers act as intermediaries, translating WebDriver commands into actions that the browser can perform. This direct communication ensures more reliable and consistent interactions compared to older methods that required a separate server.
The architecture allows WebDriver to execute commands directly in the browser, which leads to more accurate simulation of user interactions. This is in contrast to Selenium RC, which relied on injecting JavaScript into the browser to perform actions, often resulting in less stable tests. WebDriver's approach offers better control and integration with browser features, leading to a more authentic representation of user behavior.

WebDriver's functionality extends to a wide range of interactions with web elements. It can simulate user actions such as clicking buttons, entering text into fields, selecting options from dropdowns, and interacting with various elements on a web page. This capability is essential for testing the functionality and behavior of web applications, ensuring that they respond as expected under different scenarios.

1.1 Overview

This project is focused on developing an automated testing framework for the WROGN e-commerce website, with the objective of ensuring the functionality and reliability of its key features. The automation will cover various aspects of the website, including user authentication, product search, and cart management, to streamline the testing process and improve overall software quality.

The automation testing framework will be built using Selenium WebDriver, which will facilitate the simulation of user interactions such as logging in, handling OTP verification, and navigating through different sections of the site. The test scripts will automate actions like searching for products, interacting with product pages, adding items to the cart, and applying discount codes.

The project will also involve addressing dynamic elements such as pop-ups and modals, ensuring that these interactions are managed effectively within the automated tests. Comprehensive documentation will be provided for each test case, including expected and actual outcomes, to support ongoing testing and maintenance efforts. Integrating these automated tests into a Continuous Integration/Continuous Deployment (CI/CD) pipeline will enhance testing efficiency, provide rapid feedback, and ensure consistent quality in the WROGN e- commerce platform.

1.2 Objectives

1. Create a WebDriver session and Navigate to a Web page.
2. Locate the web elements on the navigated page and Perform an actions on the located elements.
3. Assert the performed actions did the correct thing and Report the results of the assertions.
4. Analyse the requirements for the given problem statement Design the solution and write test cases for the given problem.
5. Create appropriate document for the software artefact.

Chapter 2

SYSTEM SPECIFICATIONS

The system specifications for this project include a combination of hardware and software requirements designed to ensure the smooth development and execution of automated tests for the WROGN e-commerce website. The hardware should consist of a modern processor, sufficient RAM, and ample storage space, while the software needs include an up-to-date operating system, web browsers, Java Development Kit (JDK), Selenium WebDriver, and relevant development and testing tools. Detailed descriptions of the hardware and software components is provided in the following sections.

2.1 Software and Hardware Requirements

- **Hardware Requirements:**

1. Processor: Intel Core i5 or equivalent
2. RAM: 4 GB or more
3. Storage: Minimum of 2 GB free disk space
4. Screen Resolution: 1920 x 1080 (Full HD) or higher for optimal display of test results and application interactions

- **Software Requirements:**

1. Operating System: Windows 10 or higher, macOS Mojave or higher, or a recent version of Linux (e.g., Ubuntu 20.04)
2. Web Browser: Latest versions of Chrome, Firefox, Safari, or Edge.
3. Java Development Kit (JDK): Version 8 or higher.
4. Selenium WebDriver: Latest stable version compatible with the chosen browser.
5. Integrated Development Environment (IDE): Eclipse, IntelliJ IDEA, or Visual Studio Code for script development.

6. Build Automation Tool: Apache Maven or Gradle for managing project dependencies and build processes.
7. Additional Libraries: WebDriver Manager or similar for managing browser drivers automatically.

- **Network Requirements:**

1. Internet Connection: Stable and high-speed internet connection for accessing the WROGN website and downloading necessary dependencies.

- **Security and Permissions:**

1. Browser Security Settings: Ensure browser settings are configured to allow automated testing without interference from security prompts or pop-ups.
2. Firewall/Antivirus: Configure to allow communication between the automation tool and the web browser, if necessary.

Chapter 3

SYSTEM DESIGN

The block diagram for the WROGN website automation outlines a detailed sequence of steps necessary for simulating a user's shopping experience, from logging in to completing a purchase

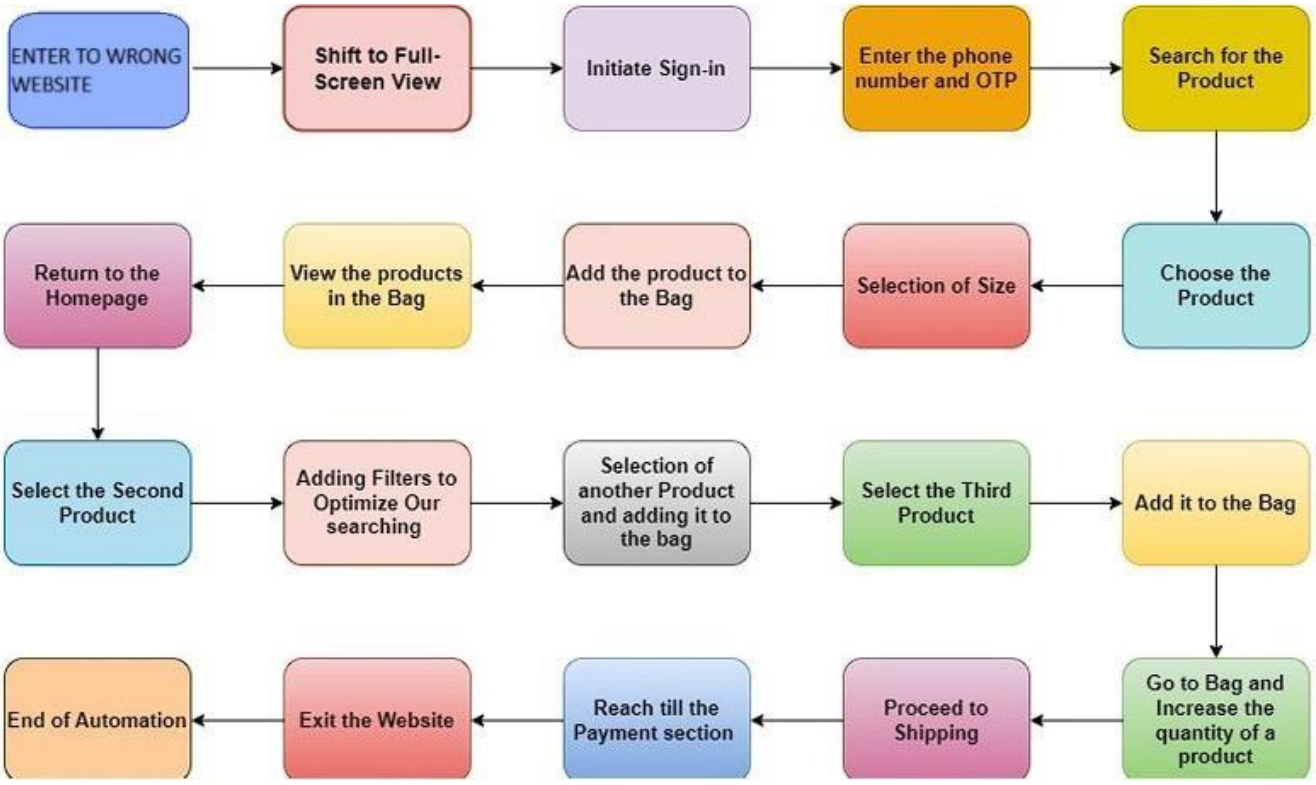

Figure 3.1 Block Diagram of steps followed

The block diagram for the WROGN website outlines a detailed user journey designed to facilitate a seamless shopping experience. The process begins with the user entering the WROGN website, which serves as the initial point of interaction. Upon entry, the user maximizes their browser to full-screen mode to ensure all elements of the website are fully visible and accessible, enhancing their overall browsing experience.

Next, if the user is not already registered, they initiate the sign-up process. This involves entering their phone number and receiving a One-Time Password (OTP) for verification. This step is crucial for verifying the user's identity and securing their account.

Once the sign-up process is complete, the user can start shopping by searching for specific products using the search bar. This search functionality is essential for helping users find items quickly within the extensive product catalog.

After conducting a search, the user selects a product from the search results. This involves clicking on the product to view more detailed information such as specifications, reviews, and pricing. If the product is size-specific, such as clothing or shoes, the user must select the appropriate size before proceeding. The selected product, along with the chosen size, is then added to the shopping bag, moving it into a virtual cart in preparation for purchase.

The user can then view all the items in their shopping bag, which is essential for reviewing selected products and ensuring they have everything they need before checking out. If the user wishes to continue shopping, they can return to the homepage and repeat the search and selection process for additional products. This involves selecting a second product, which is added to the bag, and optionally applying filters to optimize their search for more specific needs.

After adding more products to the bag, the user selects a third product, which is also added to the bag. At this point, the user may go to the bag and increase the quantity of a specific product if needed. Once satisfied with their selections, the user proceeds to the shipping section, moving closer to completing their purchase.

The next step involves reaching the payment section, where the user finalizes their order by providing payment details. Upon successful payment, the user exits the window, signifying the end of their shopping session. This comprehensive process, from entering the website to completing a purchase, is designed to be user-friendly and efficient, ensuring a positive shopping experience on the WROGN platform.

Chapter 4

IMPLEMENTATION

This chapter outlines the implementation process for the automated testing framework developed for the WROGN e-commerce website. It details the procedures for configuring the Selenium WebDriver and executing automated test scripts, as well as the interfaces of the project. The implementation ensures that the automated tests are properly set up, executed, and managed to validate the website's functionalities efficiently. The following sections describe the step-by-step procedure of the project, including the configuration of Selenium WebDriver and the execution of test scripts, followed by an overview of the main interface pages that were tested.

4.1 PROCEDURE

1. Environment Setup :

1.1. 1st method: Manual Configuration

- Step 1 : Create a Folder in C drive.
- Step 2 : Open Eclipse ; Place the path of the previously opened folder as the path and create a Java project (say by name webdriverproject)
- Step 3 : Download Selenium Web driver (Recent Version : 4.22.0) ; It will be in ZIP format, extract it into a folder.
- Step 4 : Attach the web driver libraries (JAR files) to the Java Project.
- Right Click on Java Project Name

  -> Properties

  -> Java Build Path

  -> Libraries

  -> Class Path

-> Add External Jars

-> Apply and Close (You can see Referenced Libraries under src)

1.2. 2nd method: Automated Configuration

- Step 1 : Create a simple MAVEN Project.
- Step 2 : Give GroupID and Artifact ID same name (Seleniumwebdriver) and click Finish.
- Step 3 : Click on pom.xml
- Step 4 : Add the dependencies : Search for Selenium Java in MVN repository (https://mvnrepository.com/) and click on the version needed (4.22).An XML code will be visible.Copy that code and paste it in pom.xml file between <dependencies> </dependencies> tag.
- Step 5 : Save the MAVEN Project.

  -> Right click Project name.

  -> Maven.

  -> Update Maven Project.

  -> Select Force Update of snapshots/releases.
- Once project is saved, all the jar files from web driver are saved inside Maven Dependencies automatically.

2. Some Important Definitions :

- **JAR Files:**

JAR (Java Archive) files are Java archive files that package multiple Java classes and resources into a single file.They are used to distribute Java applications or libraries in a compact, portable format.JAR files serve as a way to package Java class files, resources (such as images, configuration files) into a single file.This simplifies distribution and deployment of Java applications or libraries. JAR files use the ZIP file format, making them platform-independent and allowing easy extraction and manipulation using standard tools.

- **MAVEN Project:**

Maven is an open-source build automation and project management tool widely used for Java applications. As a build automation tool, it automates the source code compilation.A Maven project in Java refers to a software project that uses Apache Maven as its build automation and project management tool.Project Object Model (POM): Maven projects are defined by a POM file (pom.xml), which is an XML file that describes the project's configuration, dependencies, and build settings.It serves as the blueprint for how Maven builds the project.

- **Pom.xml:**

The pom.xml file in Maven is a Project Object Model (POM) file.It is an XML file that serves as the fundamental configuration file for a Maven-based project. The pom.xml file contains information of project and configuration information for the maven to build the project such as dependencies, build directory, source directory, test source directory, plugin, goals etc

- **Dependencies:**

Maven manages dependencies for a project through the pom.xml.Dependencies are specified using <dependencies> tags, where you list external libraries or modules that your project requires.Maven then automatically downloads these dependencies from repositories like https://mvnrepository.com/

3. Step by Step Procedure :

Step 1: Initialise WebDriver : Set Chrome Options and Initialise ChromeDriver:

- Configure ChromeOptions to deny location requests. This prevents location prompts during automation.
- Create an instance of ChromeDriver with the specified ChromeOptions.

Step 2: Navigate to WROGN and Perform Login

1. **Open WROGN Website :** Navigate to the WROGN homepage and maximise the browser window.

Figure 4.1 WROGN Website Homepage

2. **Click on Sign-In Button :** Use WebDriverWait to wait for the sign-in button to be clickable and then click it.

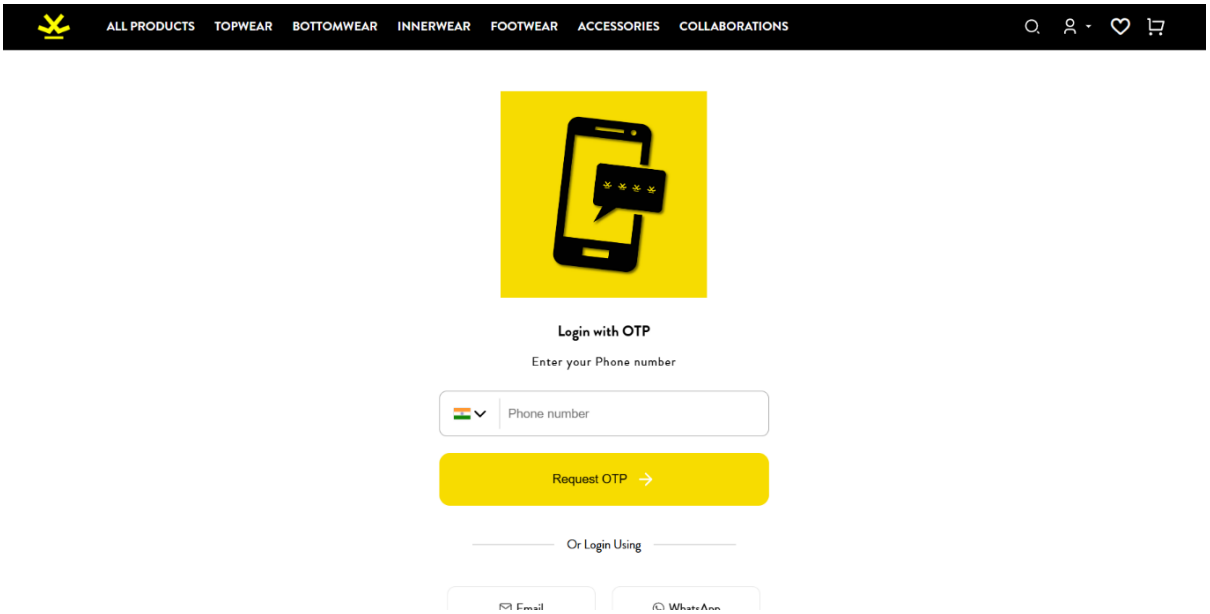

Figure 4.2 Sign-in page

3. **Enter Phone Number :** Locate the input field for the phone number and enter the test phone number.

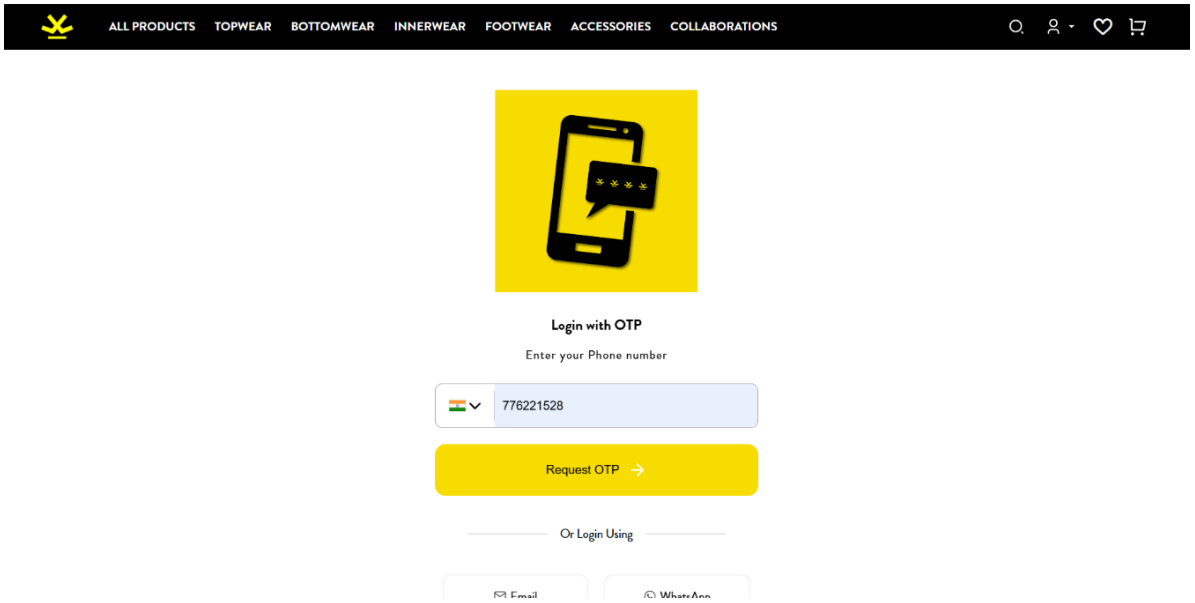

Figure 4.3 Enter login credentials

4. **Click Continue Button :** Click the continue button to proceed to OTP verification.

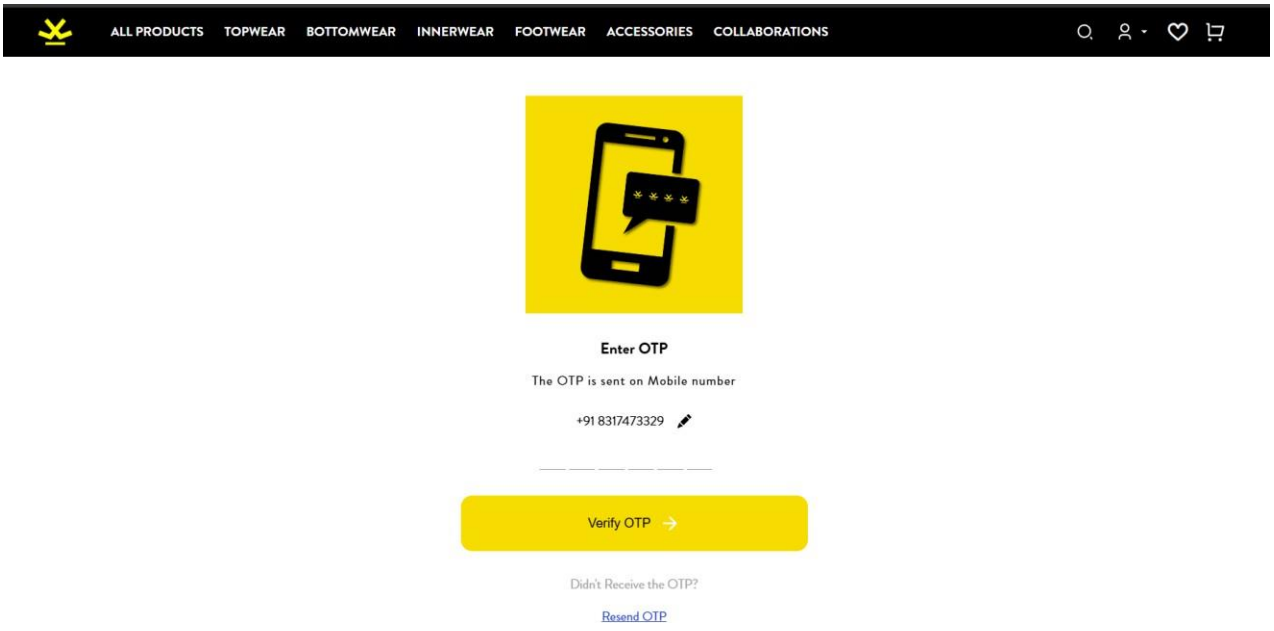

Figure 4.4 Enter OTP

5. **Invalid OTP** : If the OTP entered is Invalid then Resend OTP button is clicked.

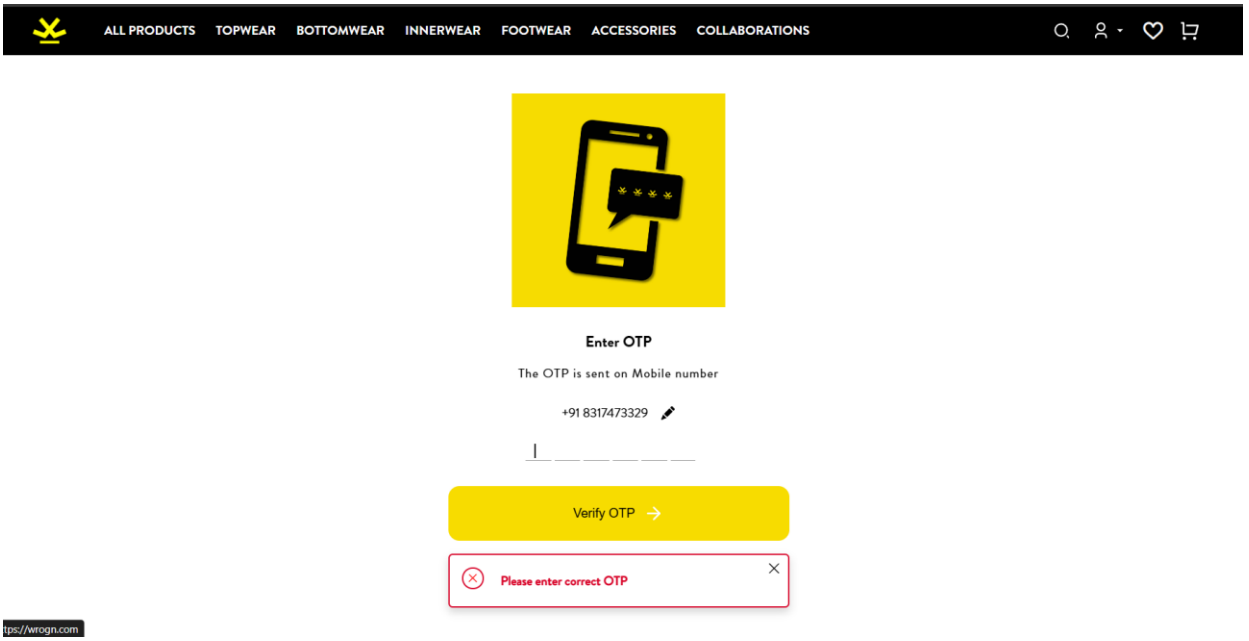

Figure 4.5 Invalid OTP Entered

6. **Valid OTP** : If Valid OTP is entered then we proceed to the home page.

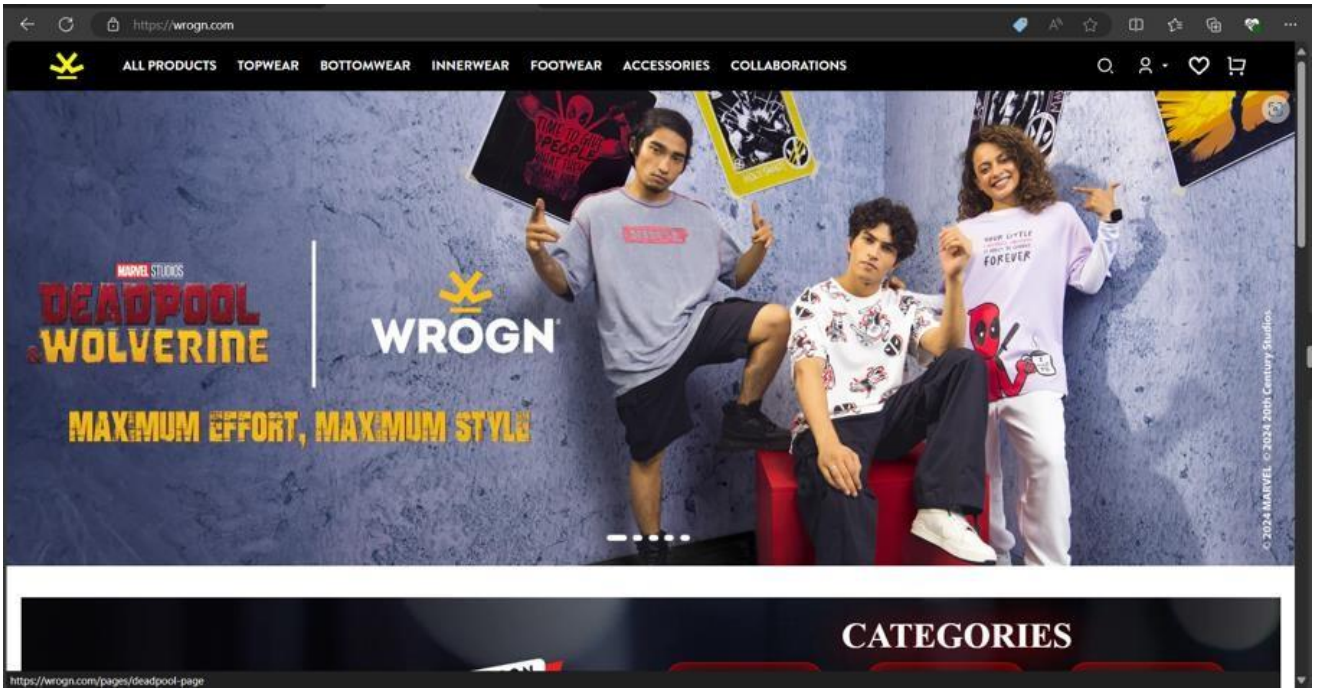

Figure 4.6 Valid OTP Entered

Step 3: Search for Products and Manage Cart

1. **Search for Products :** Enter the search term in the search box

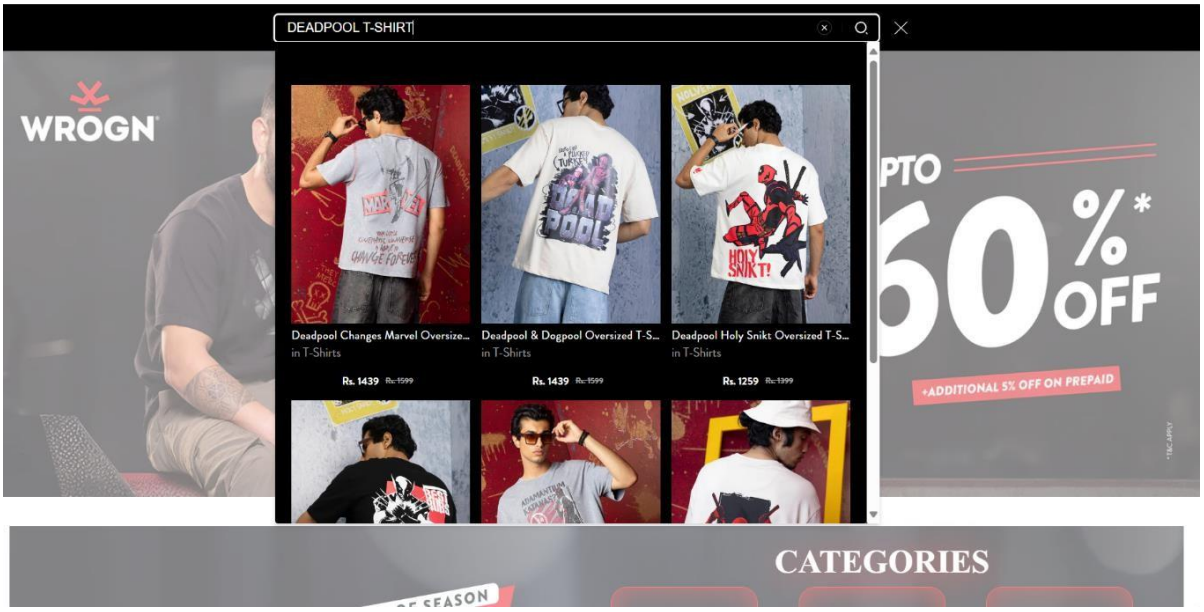

Figure 4.7 Search for 'DEADPOOL T-SHIRT'

2. **Click on Search Button**

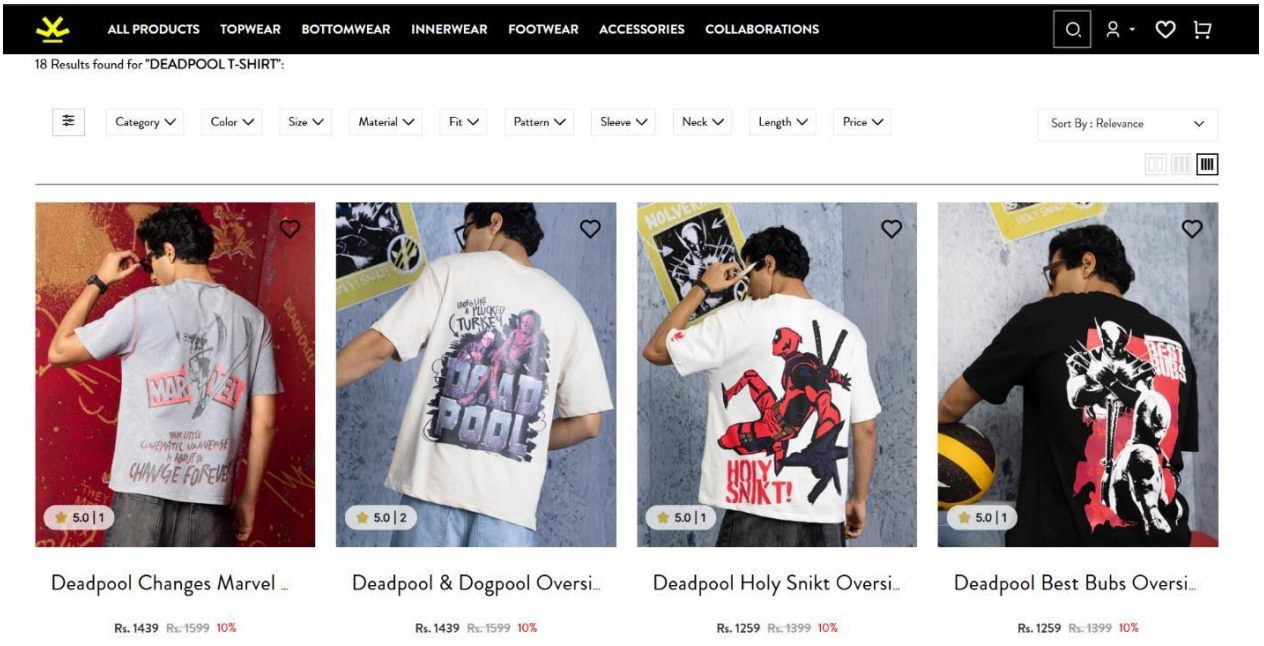

Figure 4.8 Result of the searched product

3. **Select any Product :** After selecting we will be directed to that product in another webpage. Select the required size.

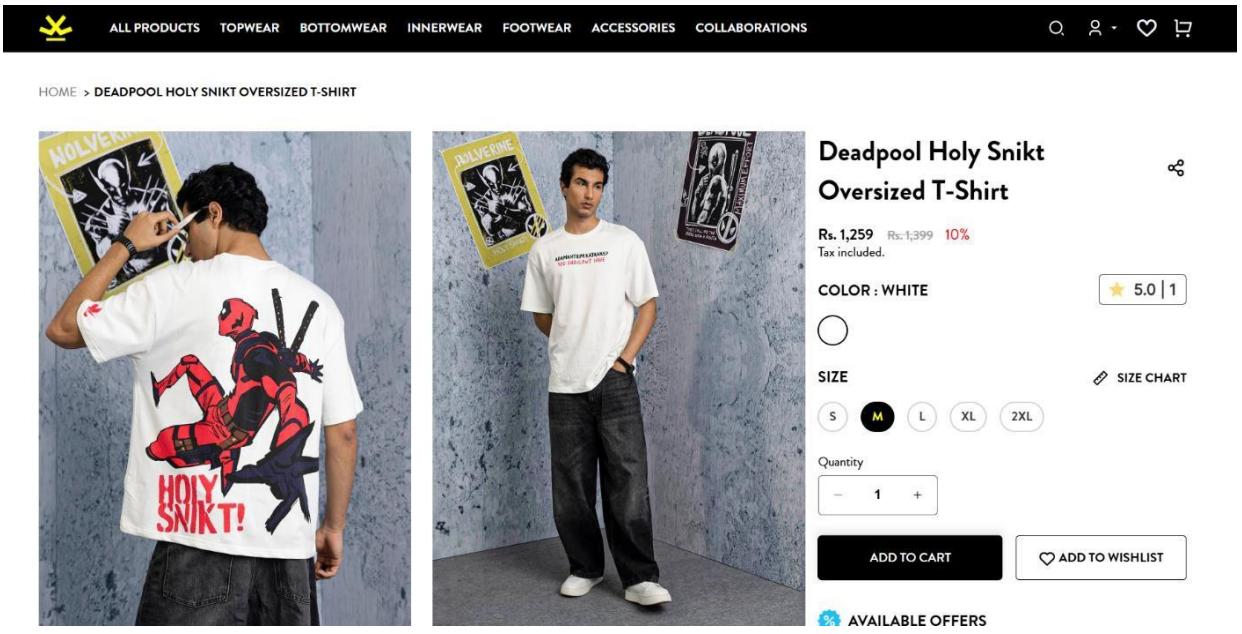

Figure 4.9 Selection of the size

4. **Add Item to Cart** : Click on Add to Bag Button and view the Cart Contents.

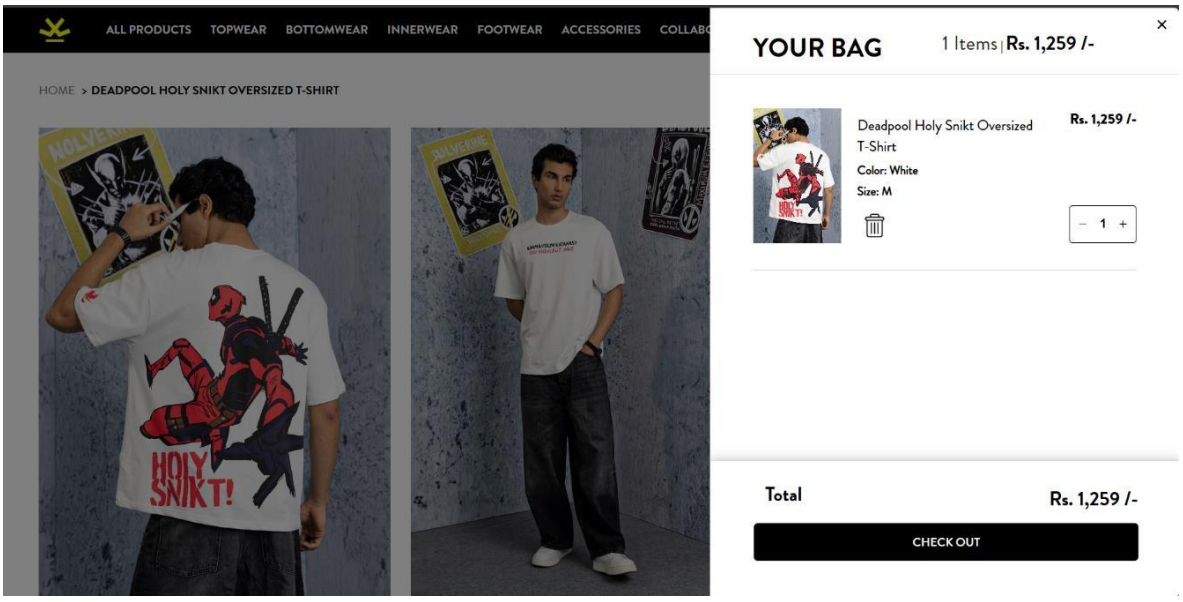

Figure 4.10 Adding product to bag

5. **Apply Coupon Code** : Apply coupon code if anything is applicable for the product selected

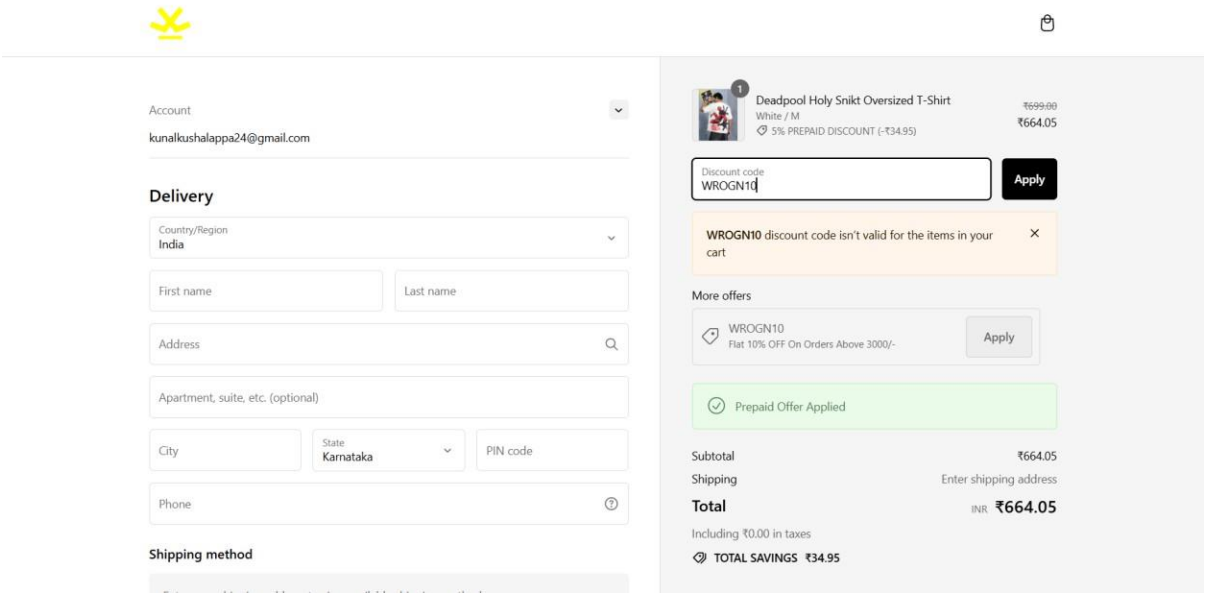

Figure 4.11 Applying coupon code

6. **Navigate back to home page** : After adding the first product to the cart ,navigate back to home page to select other products

Figure 4.12 Navigate back to homepage

7. **Select Category option and Apply Filters** : Click on the category option from the navigation bar and apply filters such as Jackets

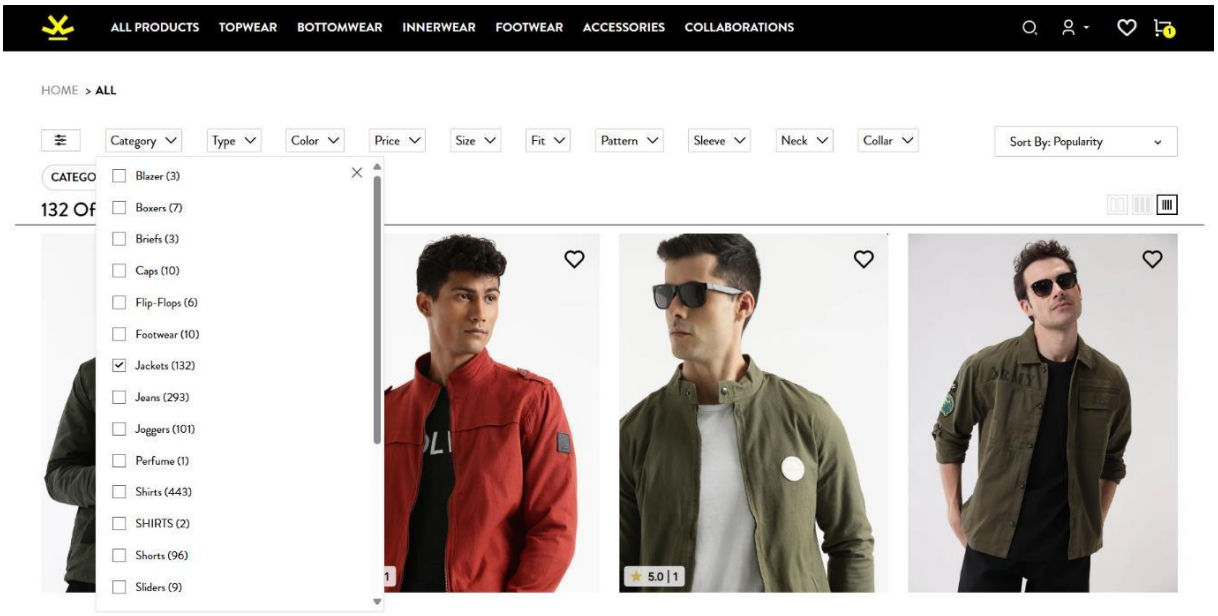

Figure 4.13 Selecting Category Option

8. **Quick View of a Product** : Hover on a product and click on the quick view option and add the product to bag. After clicking on the Add to Bag option ,the quick view will be disabled and home page will be visible.

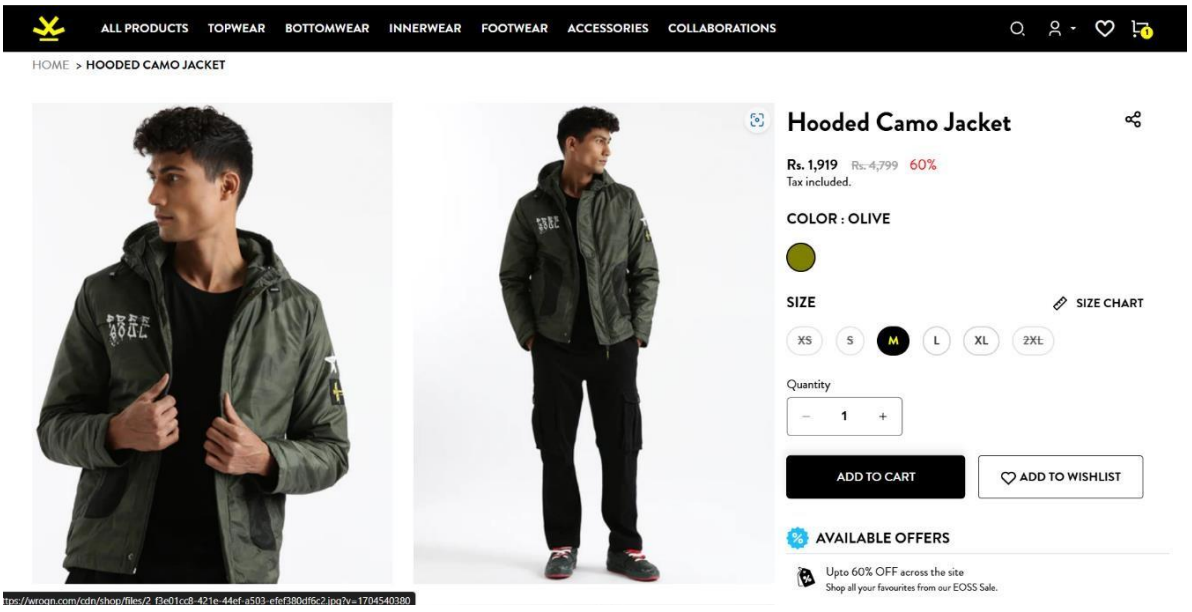

Figure 4.14 Quick View of the Product

9. **Selecting Pattern and apply filters**: Click on the filters options such as camouflage,colourblocked in the navigation bar

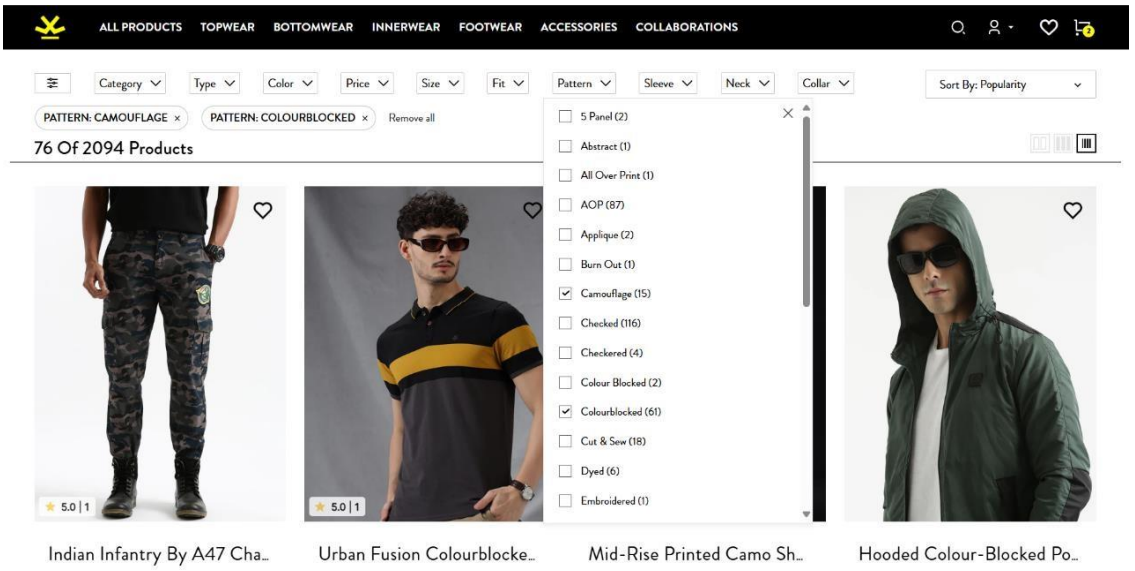

Figure 4.15 Selecting the third product

10. **Add the third item to the cart through quick view**: Similar to the previous item, again though Quick View another product will be added to cart.

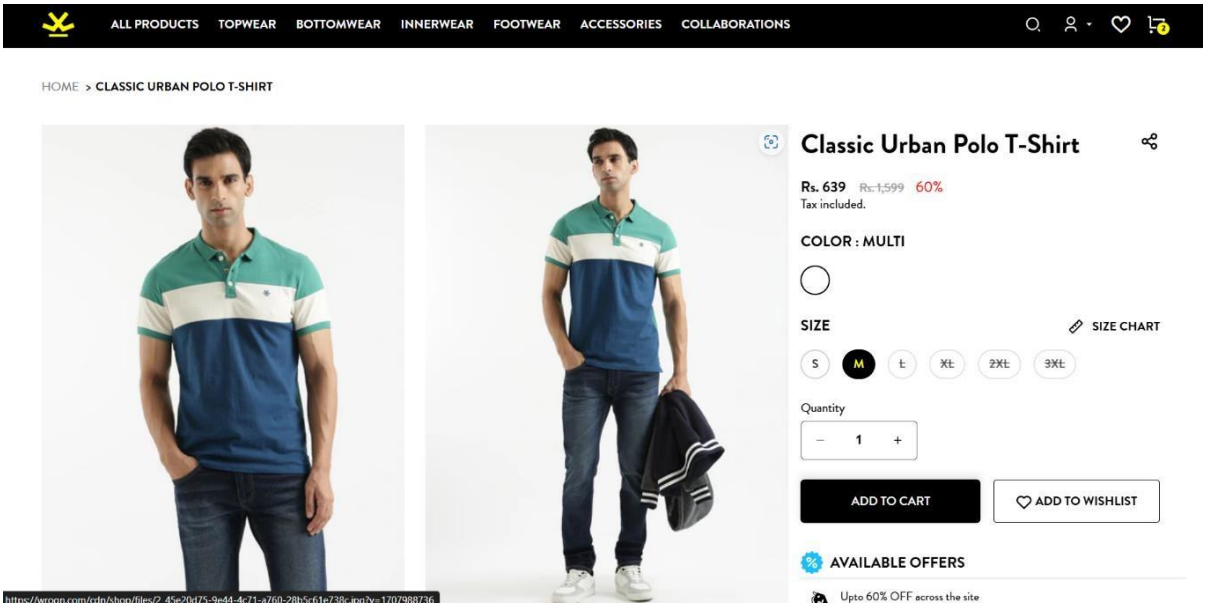

Figure 4.16 Add third item to bag

11. **Navigate to My Bag**: All the products added can be seen in My Bag.

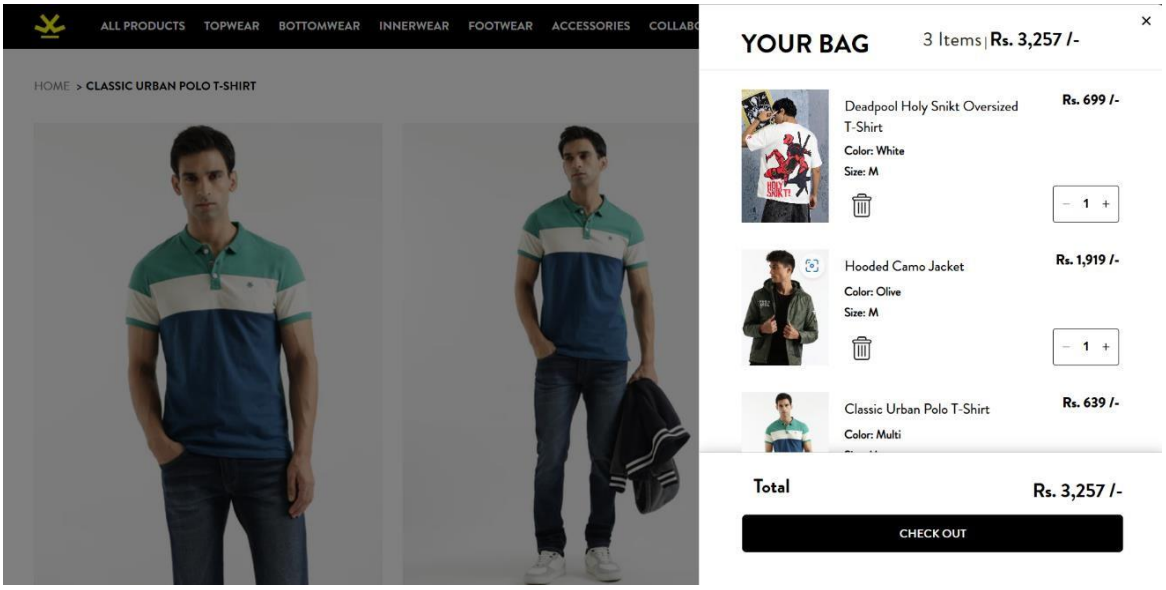

Figure 4.17 Navigate to My Bag

12. **Update Quantity of a product** : Select a product and Increment it's quantity by one.

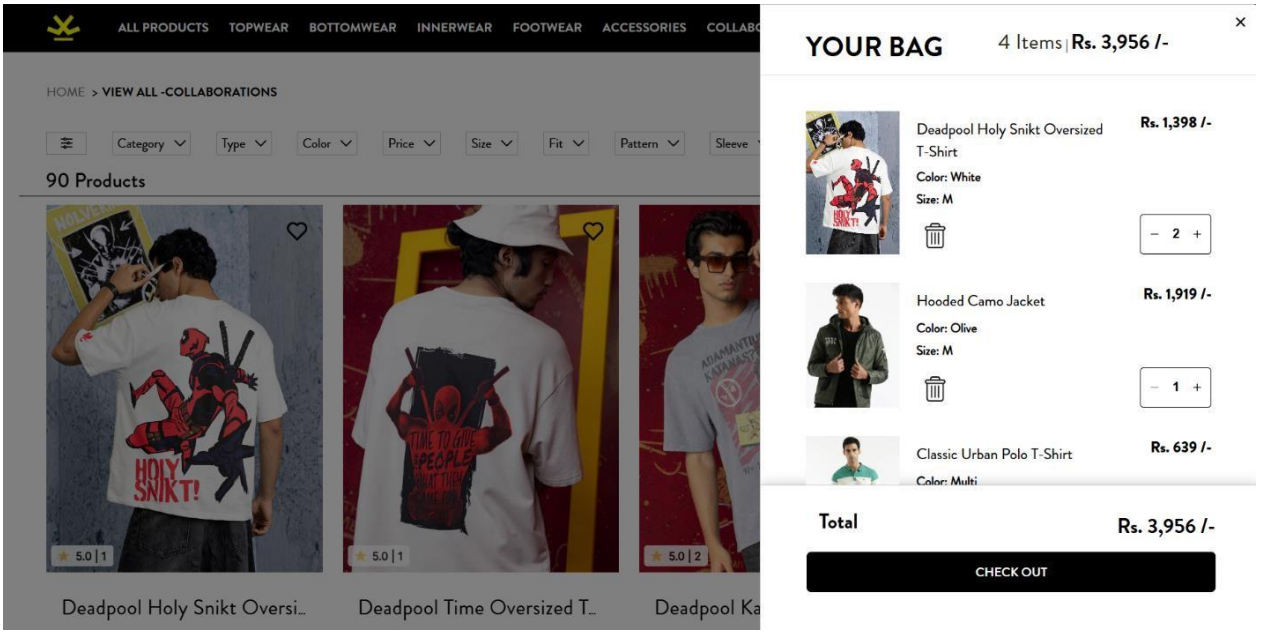

Figure 4.18 Update Item Quantity

13. **Proceed to Shipping** : We can then proceed to Shipping.

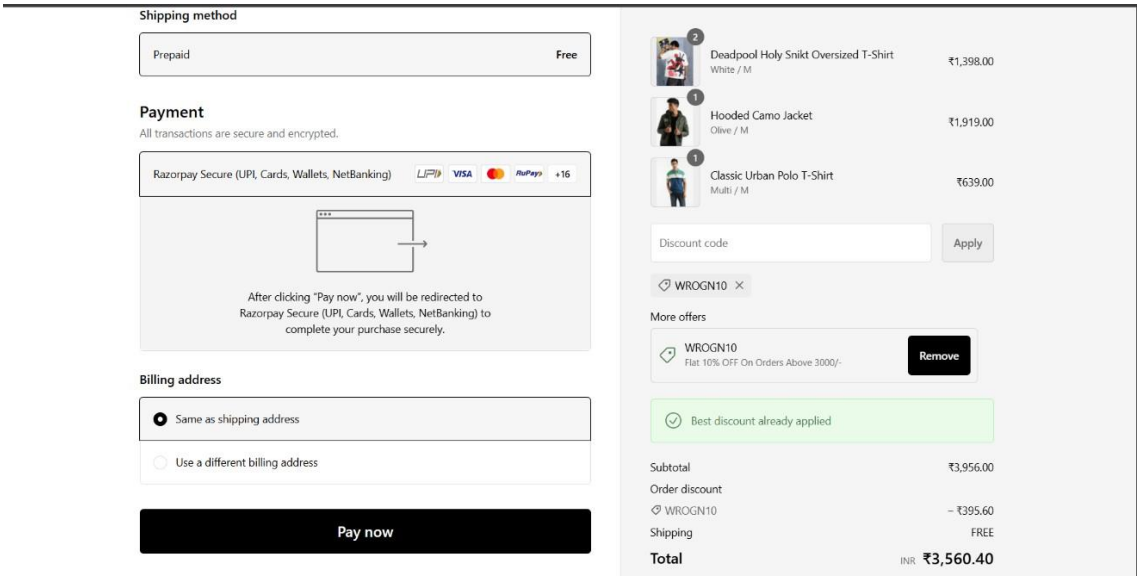

Figure 4.19 Proceed to Shipping

4.2 INTERFACES

The main interfaces in the automation process of WROGN

Figure 4.20 Home Page of WROGN

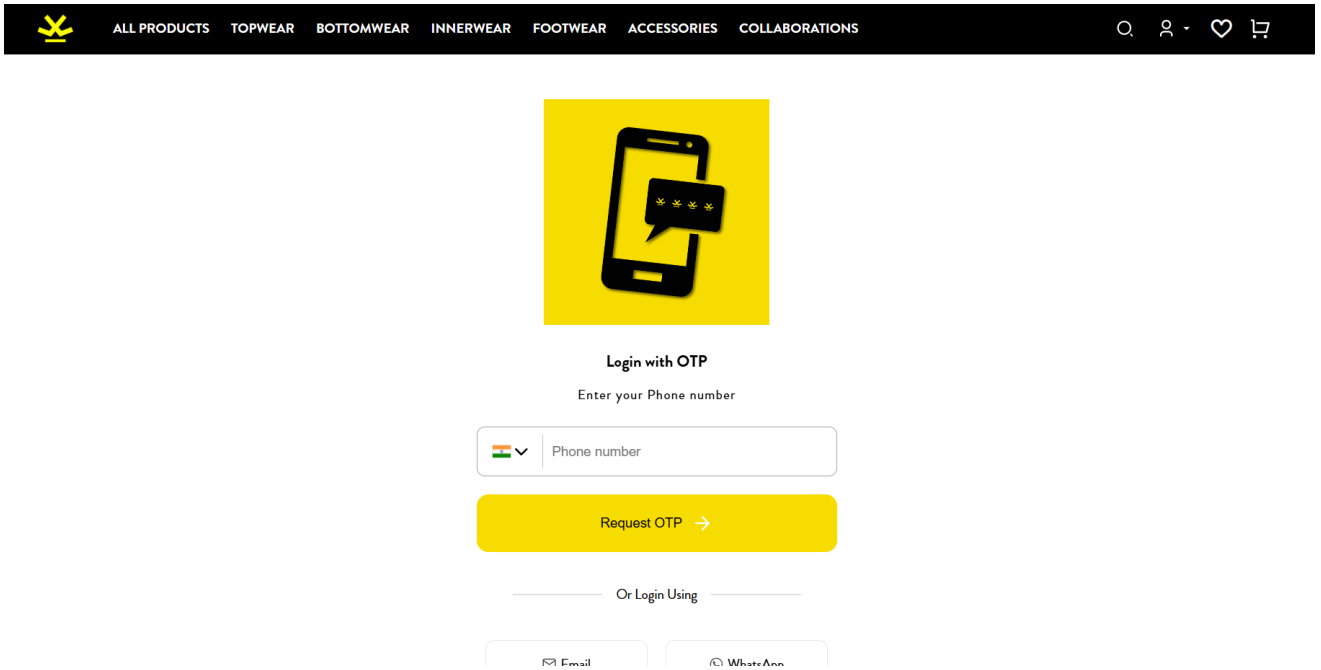

Figure 4.21 Sign-In Page

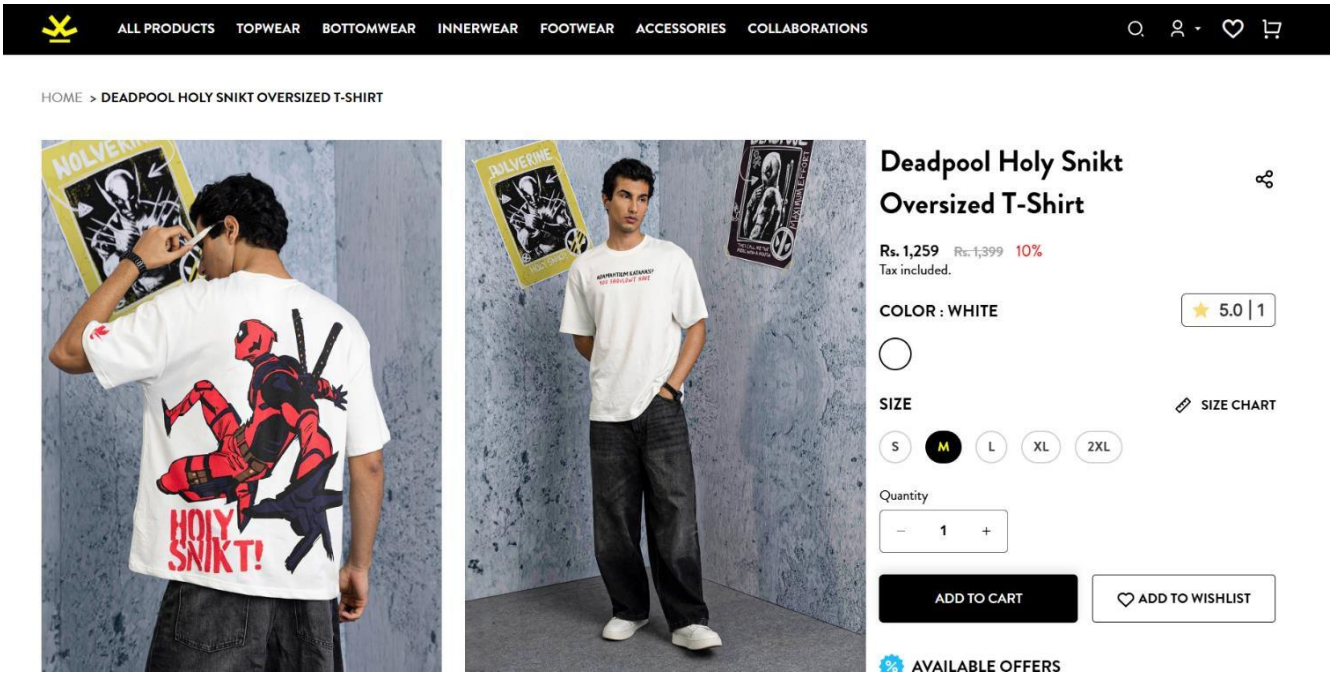

Figure 4.22 Adding the first product to cart

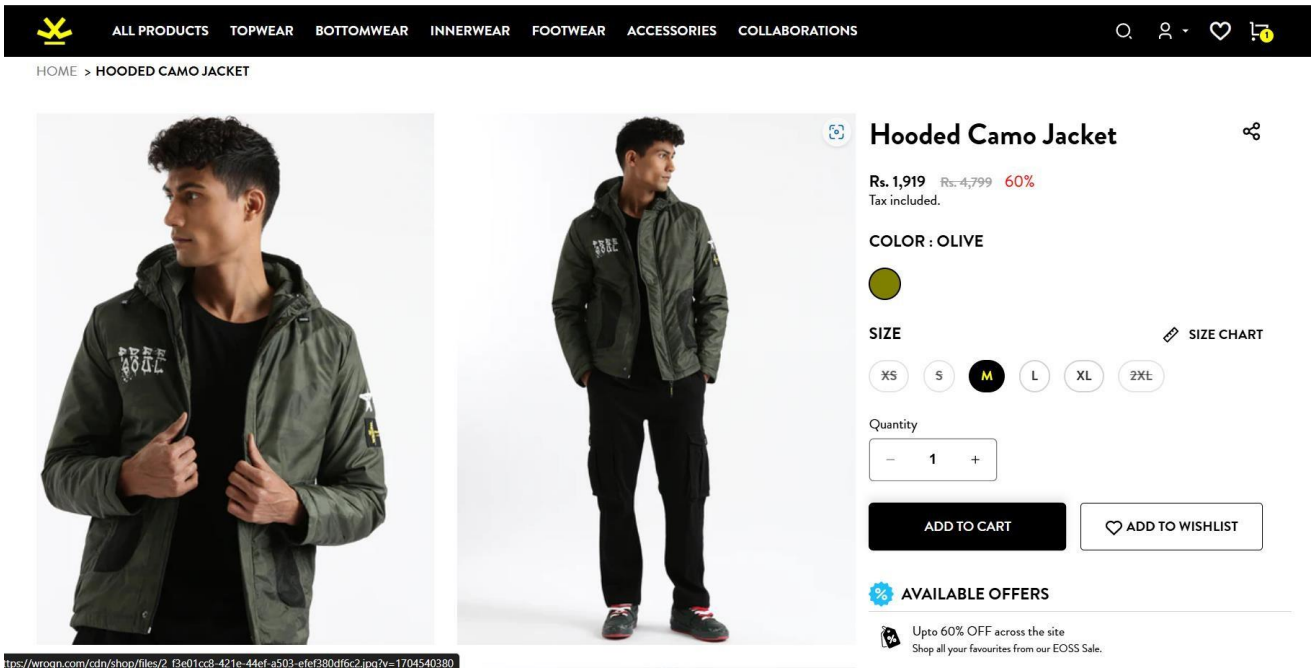

Figure 4.23 Adding the second product to cart

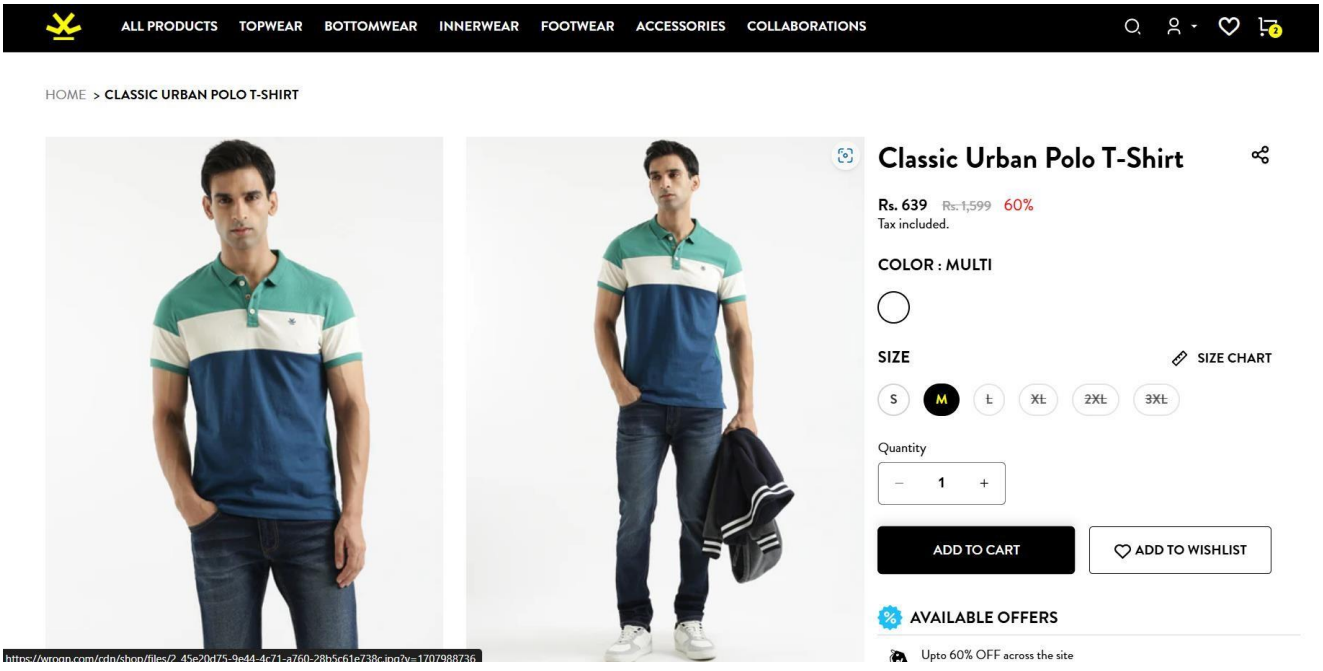

Figure 4.24 Adding the third product to cart

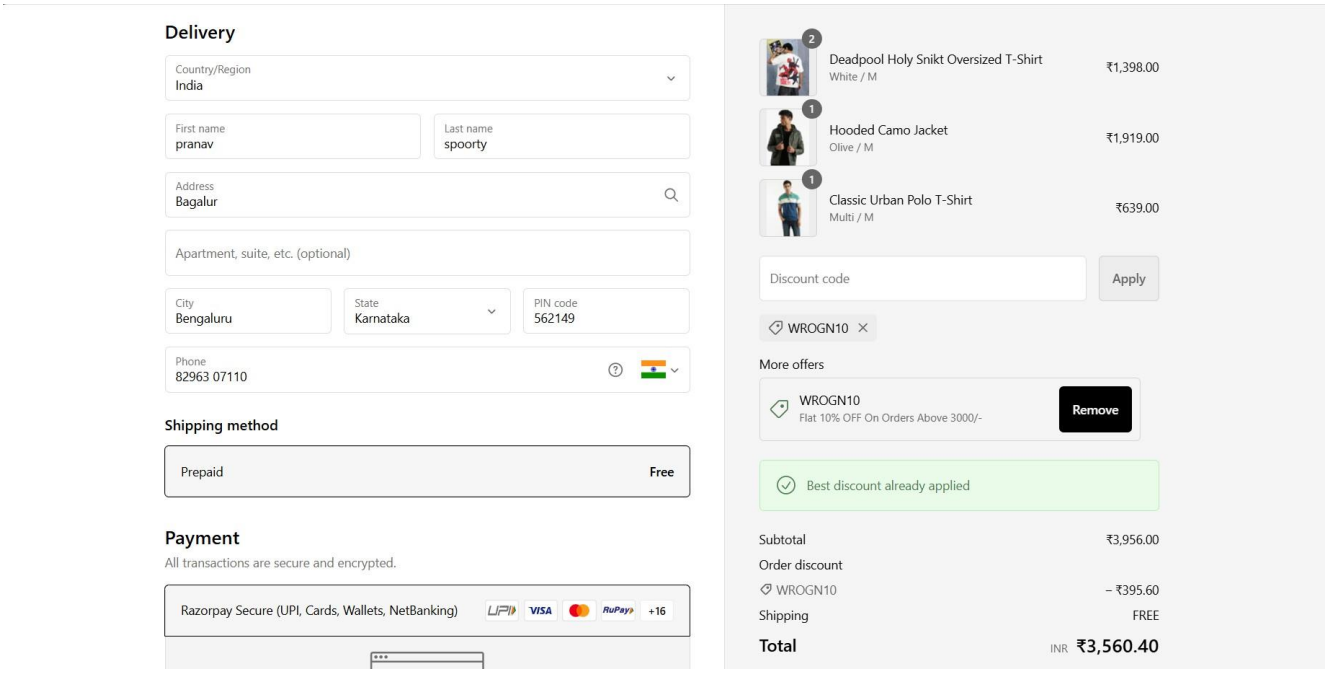

Figure 4.25 Proceed to Shipping

Chapter 5

TEST CASES

Test Case ID	Description	Steps to Execute	Expected Results	Actual Results	Status
1	Application runs	1. Run the application.	Application should open successfully.	Application opened successfully.	Pass
2	Full-screen view	1. Run the application. 2. Click on maximize screen button.	Screen should be maximized.	Screen maximized.	Pass
3	User sign-up (Invalid OTP)	1. Enter the phone number and OTP. 2. Click on the login button.	Invalid OTP message should be displayed along with Resend OTP button	Invalid OTP message displayed along with Resend OTP button	Pass
4	User sign-up (Valid OTP)	1. Enter the phone number and OTP. 2. Click on the login button.	User should be logged in.	User logged in.	Pass
5	Search functionality	1. Log in successfully. 2. Enter "Deadpool T-shirt" in the search box.	Search results should be displayed.	Search results displayed.	Pass
6	Product selection	1. Search for "Deadpool T-shirt". 2. Select a product.	Product details should be displayed.	Product details displayed.	Pass
7	Adding product to the bag	1. Select a product. 2. Click on "Add to bag" button.	Product should be added to the bag.	Product added to the bag.	Pass
8	Viewing products in the bag	1. Click on the bag icon.	Products in the bag should be displayed.	Products displayed in the bag.	Pass
9	Returning to the homepage	1. Click on the WROGN logo.	Homepage should be displayed.	Homepage displayed.	Pass

Test Case ID	Description	Steps to Execute	Expected Results	Actual Results	Status
10	Selection of second product	1. Search for a product under "Jackets" category. 2. Select a product.	Product should be selected.	Product selected.	Pass
11	Adding second product to the bag	1. Select a product. 2. Click on "Add to bag" button.	Product should be added to the bag.	Product added to the bag.	Pass
12	Applying filters	1. Navigate to a Pattern. 2. Apply filters to the search.	Filters should be applied.	Filters applied.	Pass
13	Selection of third product	1. Search for a product under "Camflouge" category. 2. Select a product.	Product should be selected.	Product selected.	Pass
14	Adding third product to the bag	1. Select a product. 2. Click on "Add to bag" button.	Product should be added to the bag.	Product added to the bag.	Pass
15	Increasing the quantity of a product	1. Go to the bag. 2. Increase the quantity of a product.	Quantity should be increased.	Quantity increased.	Pass
16	Proceeding to shipping	1. Click on the "Proceed to Shipping" button.	Shipping options should be displayed.	Shipping options displayed.	Pass
17	Reaching the payment section	1. Complete the shipping details. 2. Proceed to payment.	Payment options should be displayed.	Payment options displayed.	Pass
18	Exiting the application	1. Click on the close button.	Application should close.	Application closed.	Pass
19	End of automation	1. Complete all actions. 2. Check for completion.	Automation should end.	Automation ended.	Pass

Chapter 6

CONCLUSION

In this project, we have successfully automated the shopping process on the WROGN website using Selenium WebDriver. The automation script, written in Java, performs a series of actions that mimic a user's interaction with the website. Starting from accessing the WROGN homepage and initiating the sign-in process, the script handles user input for phone number and OTP verification, including scenarios where an invalid OTP is entered. The script then proceeds to search for specific products like Nike Jordans, add them to the shopping bag, apply necessary filters to optimise searching, and select additional products, ensuring a realistic and comprehensive shopping experience.

The automation script also incorporates functionalities like scrolling, switching between windows, handling pop-ups, and interacting with various web elements using explicit waits and JavaScript execution. This robust implementation ensures that the script can navigate through different sections of the website, add products to the cart, and proceed to the payment section. By completing these tasks, the project demonstrates the potential of Selenium in automating complex web interactions, providing a reliable framework for testing e-commerce websites. The structured flow and detailed handling of various user scenarios enhance the reliability and efficiency of the automation process, making it a valuable tool for QA engineers and developers.

Chapter 7

REFERENCES

[1] https://www.eclipse.org/downloads/

[2] https://www.selenium.dev/downloads/

[3] https://mvnrepository.com/

[4] https://www.oracle.com/in/java/technologies/

[5] https://www.wrogn.com/

[6] https://www.globalapptesting.com/blog/what-is-automation-testing

[7] https://www.geeksforgeeks.org/automation-testing-software-testing/

[8] https://selectorshub.com/

[9] https://github.com

[10] https://smartbear.com

[11] https://reintech.io/

[12] https://scalybee.com/