VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", BELAGAVI – 590018

A MINI PROJECT REPORT ON

"NOPCOMMERCE"

Submitted in partial fulfilment of requirements for the course
Software Testing Laboratory[21ISL66] of Sixth Semester of
Bachelor of Engineering in Information Science & Engineering
during the academic year 2023-24

Submitted By

DHANUSHREE CNSHAMITHA R
4MH21IS0274MH21IS082

Under the Guidance of
PROF. SHALINI M S
Professor & Head,
Dept. of ISE

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE
Belawadi, S.R.Patna (T), Mandya (D) – 571477.
2023 - 2024

MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE

Belawadi, S.R. Patna (T), Mandya (D) – 571477.

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the mini project work entitled **“NOPCOMMERCE”** a bonafide work carried out by **DHANUSHREE CN [4MH21IS027]** and **SHAMITHA R [4MH21IS082]** in partial fulfilment for the **Software Testing Laboratory (21ISL66)** prescribed by the Visvesvaraya Technological University, Belagavi during the year 2023-2024 for the sixth semester B.E in Information Science and Engineering. The mini project report has been approved as it satisfies the academic requirements.

Signature of Guide Signature of HOD
(**Prof.Shalini M S**) (**Dr. Sharath Kumar Y H**)
Professor & Head, Dept. of IS&E Professor & Head, Dept. of IS&E
MIT MysoreMIT Mysore

Name of the Examiners	**Signature with date**
1....................................... 2.......................................	....

ACKNOWLEDGEMENT

We sincerely owe our gratitude to all the persons who helped and guided us in completing this mini project work.

We are thankful to **Dr. B.G. Naresh Kumar, Principal, Maharaja Institute ofTechnology Mysore**, for having supported us in our academic endeavours.

We are extremely thankful to **Dr.Sharath Kumar Y H, Professor & Head, Department of Information Science and Engineering,** for his valuable support and his timely inquiries into the progress of the work.

We are greatly indebted to our guide**Prof. Shalini M S, Professor,Department of Information Science and Engineering,** for the consistent co-operation and support.

We are obliged to all **teaching and non-teaching staff members** of **Department of Information Science and Engineering,** for the valuable information provided by them in their respective field's. We are grateful for their co-operation during the period of our mini project.

DHANUSHREE CN (4MH21IS027)
SHAMITHA R(4MH21IS082)

ABSTRACT

It presents the automation testing of the Nopcommerceweb page using Selenium, a robust framework for web application testing. The objective of this project is to ensure the reliability and efficiency of theNopcommercewebsite's user interface and functionality by automating repetitive test cases. Selenium WebDriver is employed for browser automation, enabling the simulation of user interactions across different web browsers.

The testing suite covers essential features such as user login, product search, and checkout processes. This automation aims to streamline the testing process, reduce manual effort, and enhance test coverage, ensuring a seamless user experience on the Nopcommerceplatform. The report details the setup, execution, and results of the automated tests, demonstrating the effectiveness of Selenium in maintaining the quality of web applications.

Nopcommerce, a revolutionary digital platform, redefines the way consumers discover and experience premium lifestyle products. Launched in year, Nopcommerce connects discerning users with a curated selection of brands across categories such as beauty, wellness, and gourmet foods through its innovative mobile application.

The app leverages advanced technology to offer users a personalized sampling experience, allowing them to explore products from renowned and emerging brands before making a purchase commitment. By integrating e-commerce with experiential marketing, Nopcommerce not only enhances consumer engagement but also provides brands with a powerful channel to reach and convert affluent, quality-conscious consumers. This abstract explores Nopcommerce impact on consumer behavior, its business model, and its contribution to reshaping the digital marketplace for premium lifestyle products.

TABLE OF CONTENTS

Sl. No	CHAPTERS	Page No.
1	**INTRODUCTION**	
1.1	Aim of the Project	1
1.2	Overview of the Project	1
1.3	Outcome of the Project	1
1.4	Proposed System	2
1.5	Advantages	2
2	**SYSTEM REQUIREMENTS**	
2.1	Software Used	3
2.2	Software Description	3
3	**SYSTEM DESIGN**	
3.1	System Analysis	5
3.2	Stakeholder Analysis	6
3.3	System Architecture	6
3.4	Use Case Diagram	7
3.5	Test Planning	8

4	**IMPLEMENTATION**	
4.1	Test Case Development	10
4.2	Test Data Preparation	12
4.3	Test Environment Setup	13
4.4	Test Execution	15
4.5	Defect Reporting and Tracking	16
5	**RESULT ANALYSIS**	
5.1	Test Cases	17
5.2	Snapshots	19
6	**CONCLUSION AND FUTURE WORK**	
6.1	Conclusion	25
6.2	Future Work	25
7	**REFERENCES**	
7.1	Textbooks	26
7.2	Links	26

Chapter 01

INTRODUCTION

1.1 Aim of the Project:

Street style store websites aim to capture the essence of urban fashion by offering a curated selection of trendy and edgy clothing that reflects the latest styles seen on city streets worldwide. These platforms cater to fashion-forward individuals who seek to express their individuality through unique and eclectic pieces. By showcasing a diverse range of apparel and accessories, from casual basics to statement pieces, street style stores provide customers with the tools to create distinctive and contemporary looks.

1.2 Overview of the Project:

Street style store websites are dynamic platforms that cater to fashion enthusiasts looking to embrace urban trends and express their individuality through clothing. These websites typically offer a wide array of apparel, footwear, and accessories curated to reflect current street fashion trends from around the globe. They often feature a blend of established and emerging brands, showcasing a mix of casual basics, statement pieces, and niche items that appeal to a diverse customer base.

1.3Outcome of the Project:

Street Style Store, an online fashion destination renowned for its edgy and trend-setting clothing, has garnered a loyal following among fashion enthusiasts worldwide. Boasting a diverse range of apparel, from chic urban wear to statement accessories, the website has carved out a niche for itself in the competitive online retail market. Its success lies in a seamless blend of affordability and style, catering to young adults who seek to express their individuality through fashion.

1.4Proposed System:

The proposed system for the Street Style Store website aims to enhance user experience and operational efficiency through several key features. Firstly, a responsive and intuitive interface will be implemented to ensure seamless navigation and accessibility across devices. Secondly, an advanced search and filtering mechanism will allow customers to quickly find products based on preferences such as style, size, and color.

Integration with secure payment gateways will offer a smooth checkout process, bolstered by robust cybersecurity measures to safeguard customer data. Additionally, personalized recommendations based on past purchases and browsing history will enhance the shopping experience, encouraging repeat visits and increased customer satisfaction. Behind the scenes, an efficient inventory management system will optimize stock levels and ensure timely restocking of popular items, minimizing out-of-stock situations. Finally, regular updates showcasing the latest fashion trends and exclusive collections will keep the website dynamic and engaging, reinforcing Street Style Store's reputation as a trendsetter in the online fashion retail industry.

1.5Advantages:

- Trendsetting Collections
- Affordability
- Wide Range of Products
- User-Friendly Interface
- Responsive Design
- Customer Engagement
- Fast Shipping and Returns
- Exclusive Offers and Discounts

Chapter 02

SYSTEM REQUIREMENTS

2.1Software Used:

- Operating system: Windows 10 or 11.
- Front end: HTML and CSS, Java Script and Selenium WebDriver.
- Back end: JDK, Eclipse IDE, Selenium WebDriver, Apache Maven and TestNG.

2.2 Software Description:

2.2.1 ECLIPSE IDE:

Eclipse IDE, developed by the Eclipse Foundation, is an open-source integrated development environment renowned for its versatility and robust toolset across multiple programming languages. It provides developers with a comprehensive suite of features designed to enhance productivity throughout the software development lifecycle. create, copy, drop, rename and alter databases, tables, columns and indexes.

- maintenance server with proposals on server configuration.
- execute, edit and bookmark.
- load text files.
- import data from Open Document
- administer multiple servers.
- search globally in a database or a subset of it.
- transform stored data into any format using a set of predefined functions, like displaying BLOB-data as image or download-link.
- track changes on data.
- Code Editing**:** Includes features like syntax highlighting, code completion, and automatic formatting to streamline coding tasks. Support mysqli, the improved MySQL extension.
- create, edit, call, export and drop stored procedures and functions.

- create, edit, export and drop events and triggers.
- synchronize two databases residing on the same as well as remote servers.

2.2.2 JAVA:

Platform Independence: Java programs run on any device with a Java Virtual Machine (JVM).

- To support uploading of ZIP files, you need the JAVA zip extension.
- For proper support of multibyte strings (eg. UTF-8, which is currently the default), you should install the selenium extensions.
- Variables must be declared with specific data types, catching errors early.
- When using the "cookie" authenticated method, the extension is strongly suggested for most users and is required for 64–bit machines.

2.2.3 VISUAL STUDIO CODE:

Visual Studio Code (famously known as **VS Code**) is a free open source text editor by Microsoft. VS Code is available for Windows, Linux, and macOS. Although the editor is relatively lightweight, it includes some powerful features that have made VS Code one of the most popular development environment tools in recent times.

VS Code supports a wide array of programming languages from Java, C++, and Python to CSS, Go, and Dockerfile. Moreover, VS Code allows you to add on and even creating new extensions including code linters, debuggers, and cloud and web development support.

Chapter 03

SYSTEM ANALYSIS AND DESIGN

3.1System Analysis:

System analysis is a detailed of the various operations performed by a system and their relationship within and outside the system. It is a systematic technique that defines goals and objectives. The goal of system development is to develop a system in line with the user requirement, and analysis of the system plays important role. One of the main aspects of analysis is the defining the boundaries of the system.

The various tools of structured analysis are:

- ➢ Stakeholder Analysis
- ➢ System Architecture
- ➢ Use-case diagram
- ➢ Test Planning

The structured analysis has the following attributes:

- Comprehensiveness**:** Ensures all functional and non-functional requirements are covered.
- Clarity**:** Provides clear documentation and diagrams for understanding system interactions.
- Stakeholder Involvement**:** Engages all relevant stakeholders to gather comprehensive requirements and feedback.
- Scalability**:** Considers the system's ability to handle growth and increased user demand.
- Maintenance**:** Simplifies system maintenance and updates through modular and documented approaches.
- Communication**:** Enhances communication among stakeholders by using standardized models and documentation.

3.2 Stake Holder Analysis:

- Developers**:** Write and maintain the codebase; ensure functionality and maintainability.

- Testers**:** Create and execute test cases; ensure features work as intended and are bug-free.
- Project Managers**:** Oversee project progress; ensure timely completion and quality standards.
- End-Users**:** Use the platform; seek a seamless, intuitive, and secure shopping experience.
- Business Analysts**:** Gather and analyse requirements; ensure features align with business goals.

3.3 System Architecture:

- Frontend: Built with HTML, CSS, and JavaScript to provide a responsive and interactive user interface for browsing products and managing shopping carts.
- Backend**:** Utilizes Java and Spring Boot to handle business logic, process user requests, and manage data transactions between the frontend and database.
- Database**:** Employs MySQL or PostgreSQL to securely store user data, product information, order details, and transaction histories.
- External Integrations**:** Integrates with payment gateways like Stripe and PayPal to securely process financial transactions and payments.
- Selenium Testing Framework**:** Uses Selenium WebDriver and TestNG/JUnit to automate testing of frontend and backend functionalities, ensuring the application operates as expected.

System architecture is a critical aspect of software testingand system engineering, defining the structure, behavior, and deployment of complex systems to meet functional and non-functional requirements effectively. The system architecture of our website integrates a user-friendly front-end with robust back-end services, ensuring seamless interactions between the client interface and server-side logic. This setup supports efficient data management, secure transactions, and scalable performance for optimal user experience.

3.4Use Case Diagram:

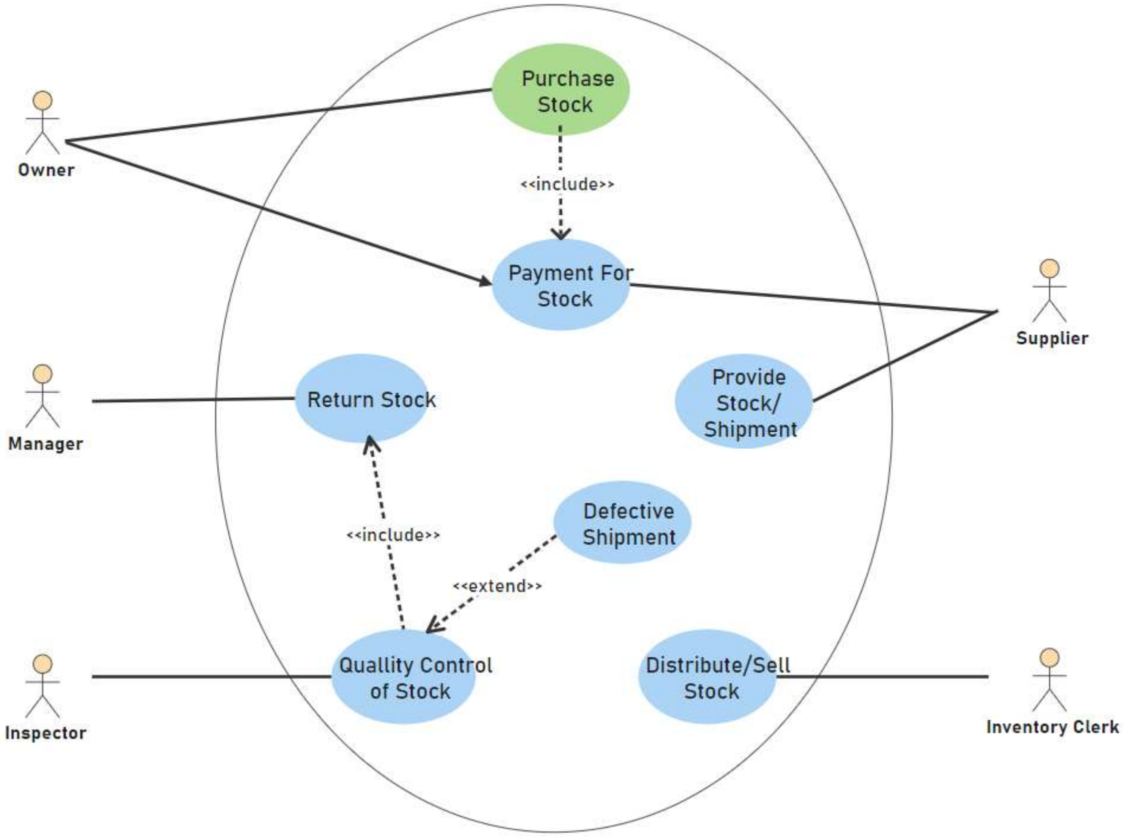


Fig.3.4 Use case diagram of Customer and Seller

- **View Product Details:** Enables users to see detailed information about a specific product, including images, descriptions, reviews, etc.
- **Add to Cart:** Allows users to add products to their shopping cart for future purchase.
- **Place Order:** Initiates the checkout process where users enter shipping details, select payment methods, and confirm their purchase.
- **Manage Products:** For administrators, includes tasks such as adding new products, updating product information, managing inventory, etc.

3.5Test Planning:

Planning a successful street style store involves meticulous attention to several key elements to ensure a vibrant and appealing retail environment. Firstly, location is paramount. Identifying a bustling urban area or a trendy neighborhood frequented by the target demographic is essential. Next, the store layout should be inviting and reflect the edgy and dynamic essence of street fashion. This could include industrial-inspired decor, flexible display areas, and ample space for customers to explore freely.

Moreover, curating the right mix of merchandise is crucial. This involves sourcing unique and on-trend clothing, footwear, and accessories that resonate with the fashion-forward clientele. Collaborations with local designers or streetwear brands can add an exclusive touch.

Additionally, integrating technology for an enhanced shopping experience, such as interactive displays or a user-friendly online store complemented by in-store pickup options, can attract tech-savvy consumers.

Lastly, effective marketing strategies like social media campaigns, influencer collaborations, and community events can build buzz and drive foot traffic. By combining these elements with a clear brand identity and excellent customer service, a street style store can carve out a niche and thrive in the competitive retail landscape.

Scope

- **E-commerce Platform**:Provide a robust platform for online shopping, showcasing a wide range of streetwear fashion items including clothing, footwear, and accessories.
- **Product Catalog**:Organize products into categories (e.g., tops, bottoms, shoes, hats) with detailed descriptions, sizing options, and high-quality images to facilitate easy browsing and purchasing.
- **User Experience**:Focus on a user-friendly interface with intuitive navigation, quick loading times, and a responsive design that works well across different devices (desktop, mobile, tablets).
- **Payment Options**:Support secure payment gateways that accept major credit/debit cards, digital wallets, and possibly alternative payment methods to cater to diverse customer preferences.
- **Customer Accounts**:Allow customers to create accounts for personalized shopping experiences, order tracking, and easy reordering.

- **Test Strategy:**
  1. **Functional Testing:**Test all website functionalities such as navigation, search, product browsing, filtering, sorting, and checkout process.
  2. **Usability Testing**:Conduct usability tests to ensure the website is intuitive and easy to navigate for different user personas.
  3. **Performance Testing**:Perform load testing to evaluate how the website handles simultaneous user interactions, especially during peak traffic periods.
  4. **Security Testing**:Conduct security assessments to identify and mitigate vulnerabilities such as cross-site scripting (XSS), SQL injection, and insecure authentication.
  5. **Compatibility Testing**:Validate website compatibility across different web browsers (e.g., Chrome, Firefox, Safari, Edge) and versions.
  6. **Integration Testing**:Verify integration with third-party services such as payment gateways, shipping providers, and analytics tools.

Test Resources:

- Tools: Selenium WebDriver, TestNG, Maven/Gradle.
- Environment: Set up test environments that mirror production settings for accurate results.

Test Deliverables:

- Test Cases: Documented scenarios for each feature and functionality.
- Test Reports: Detailed reports on test execution, including pass/fail status, issues found, and screenshots.
- Bug Reports: Document and track any defects or issues identified during testing.

Timeline:

Launching a street style store website involves several sequential phases typically spanning 4-6 months. Initially, 1-2 months are dedicated to market research, defining target demographics, and establishing unique selling propositions.

Concurrently, brand development commences, encompassing logo design, brand messaging, and visual identity creation. Product selection and sourcing follow, ensuring a curated inventory aligns with market demands and quality standards. The subsequent 2-3 months focus on website development, including design, e-commerce functionality setup, and content creation like SEO-optimized product descriptions and multimedia assets.

The final month leading to launch is crucial for rigorous testing, ensuring website functionality and user experience meet expectations.

Tools and Technologies:

Creating a street style store website requires a comprehensive set of tools and technologies to ensure a seamless user experience and efficient management. Typically, e-commerce platforms like Shopify, Magento, or WooCommerce serve as the foundation, providing essential features for product management, shopping cart functionality, and secure checkout processes.

These platforms are often complemented by content management systems like WordPress, which offer flexibility in design and content updates. Front-end development utilizes HTML, CSS, and JavaScript to create visually appealing and responsive designs, often supported by frameworks like Bootstrap or Foundation. Backend development involves languages such as PHP, Node.js, or Python (with frameworks like Django), along with databases like MySQL or MongoDB for storing and managing product information and user data securely.

Hosting on platforms like AWS or Google Cloud ensures scalability and reliability, while security measures like SSL certificates and PCI compliance safeguard user transactions. Analytics tools like Google Analytics provide insights into website traffic and user behavior, while SEO tools optimize content for search engines.

Design tools such as Adobe Photoshop or Figma assist in creating and editing visual assets, while customer support tools and email marketing platforms facilitate engagement and communication with customers.

Regular maintenance and backup procedures ensure the website runs smoothly and securely over time. By integrating these tools effectively, street style store websites can deliver a compelling shopping experience that meets modern consumer expectations.

Chapter 04

IMPLEMENTATION

4.1 Test Case Development:

Test case development is a crucial phase in the software testing lifecycle, ensuring that the system meets specified requirements and performs as expected. It involves creating a set of conditions or variables under which a tester determines whether an application or system is working correctly.

Steps in Test Case Development:

1. Requirement Analysis:
   - Review and understand the software requirements.
   - Identify testable requirements and define clear objectives for each test case.
2. Test Case Design:
   - Test Case ID: Assign a unique identifier to each test case.
   - Test Description: Provide a brief summary of what the test case will validate.
   - Preconditions: Specify any preconditions that must be met before executing the test case.
   - Test Steps: Outline the specific steps to be followed during the test execution.
   - Test Data: Define the data required for testing, including inputs and expected outputs.
   - Expected Result: Describe the expected outcome of the test case.
   - Actual Result: Record the actual result after test execution.
   - Status: Indicate whether the test case passed or failed.
3. Test Case Prioritization:
   - Categorize test cases based on their importance and impact on the system.
   - Prioritize critical and high-risk areas to ensure they are tested first.
4. Review and Validation:

- Peer review test cases to ensure accuracy and completeness.
- Validate that test cases align with requirements and cover all functional aspects.

5. Test Case Execution:

- Execute test cases in the specified environment.
- Record results, capture screenshots, and log any defects encountered

6. Maintenance:

- Update test cases to reflect changes in requirements or functionalities.
- Archive obsolete test cases and create new ones for newly added features.

Best Practices:

- Clarity and Precision: Write test cases in a clear and concise manner to avoid ambiguity.
- Reusability: Design test cases that can be reused in different testing cycles or for similar applications.
- Traceability: Ensure each test case can be traced back to specific requirements, providing a clear linkage between requirements and test cases.
- Automation: Where possible, automate repetitive and regression test cases to save time and effort.

By developing well-structured and detailed test cases, the testing process becomes more organized, efficient, and effective, ensuring a higher quality software product.

4.2 Test Data Preparation:

Preparing test data for a street style store involves several key steps to ensure the collection is comprehensive and representative of your target audience. Firstly, gather a diverse range of clothing items that reflect current fashion trends, including casual wear, urban streetwear, and statement pieces. It's crucial to include various sizes to cater to different body types and preferences.

Certainly! Here are the steps in test data preparation for a street style store:

Steps in Test Data Preparation:

1. **Define Test Objectives**:Determine the purpose of the test data. Are you testing a new collection, assessing customer preferences, or evaluating sales potential?
2. **Select Clothing Items**:Choose a diverse range of clothing items that represent your store's style. Include tops, bottoms, dresses, outerwear, and accessories. Ensure you have various sizes to cater to different body types.
3. **Organize Categories**:Categorize the selected items logically (e.g., by type, seasonality) to streamline testing and analysis.
4. **Prepare Product Information**:Gather detailed product information for each item, including materials, sizes available, care instructions, and any unique features.
5. **Photograph Items**:Take high-quality photos of each item from multiple angles. Use consistent lighting and backgrounds to maintain a professional look.
6. **Write Descriptions**:Create informative and engaging descriptions for each product. Highlight key features, style suggestions, and any other relevant details to help customers make informed decisions.
7. **Catalog and Document**:Organize all information, photos, and descriptions in a structured manner. This could be in a spreadsheet, database, or dedicated software.
8. **Review and Quality Check**:Ensure all data is accurate and complete. Review photos for clarity and consistency, and verify product information.
9. **Upload to Test Environment**:If applicable, upload the test data to your website's test environment or any other platform you are using for testing purposes.
10. **Monitor Performance**:Track how customers interact with the test data. Analyze metrics such as views, click-through rates, and conversion rates to assess the effectiveness of the test data set.

4.3 Test Environment Setup:

Setting up a test environment for a street style store website involves creating a replica of the production environment where you can safely test changes and updates without affecting the live site. Here's a step-by-step guide to setting up such an environment:

1. Choose a Development Environment:
   - **Local Development:** Set up a development environment on your local machine using tools like XAMPP, WAMP, MAMP (for Windows, macOS, and Linux respectively).
   - **Cloud Development:** Use cloud-based development environments like AWS Cloud9, Gitpod, or similar services.
   - **Virtual Machines/Containers:** Use tools like Docker to create isolated containers or virtual machines to mimic your production environment.
2. Clone the Website:
   - Clone the repository of your street style store website from your version control system (e.g., Git repository). Ensure you have the latest codebase.
3. Set Up the Database:
   - **Export Database:** Export a copy of your production database (if permissible) or use sanitized sample data.
   - **Create a Local Database:** Set up a local database (e.g., MySQL, PostgreSQL) and import the database dump.
4. Configure Environment Variables:
   - Set up environment variables for your development environment. This might include database connection details, API keys, etc. Use dotenv (.env) files for managing sensitive information securely.
5. **Install Dependencies:**
   - If your project uses package managers like npm (for Node.js projects) or Composer (for PHP projects), install dependencies required for the project.
6. Adjust Configuration:
   - Modify configuration files (e.g., `config.php`, `.env`) to reflect your local environment settings (e.g., database credentials, base URLs).

4.4 Test Execution:

Testing a street style store website involves verifying various aspects to ensure it functions correctly, is user-friendly, and meets business requirements. Here's a structured approach to test execution:

- Functionality Testing:
  - Navigation:
    - Verify all navigation links (Home, Shop, Categories, About Us, Contact) lead to the correct pages.
    - Test dropdown menus (if any) to ensure they appear and function properly.
  - Search Functionality:
    - Enter different keywords in the search bar and verify results are accurate.
    - Test advanced search options (if available) such as filtering by category, price range, etc.
  - Product Listings:
    - Ensure products are displayed correctly with accurate prices, descriptions, and images.
    - Verify sorting options (e.g., by price, popularity) work as expected.
- User Experience Testing:
  - Responsive Design:
    - Test website responsiveness on different devices (desktop, mobile, tablet).
    - Ensure elements resize and rearrange properly for each screen size.
- Performance Testing:
  - Page Load Speed:
    - Measure and optimize page load times using tools like Google PageSpeed Insights.
    - Ensure images and content load quickly across different pages.
- Security Testing:
  - SSL Certificate:
    - Verify the website uses HTTPS and SSL certificates are valid and up to date.

        - Check for mixed content warnings that could affect security.

- Content Management and SEO:
    - Content Accuracy:
        - Ensure product descriptions, prices, and availability are accurate and up to date.
        - Check for broken links and outdated content.
- Integration Testing:
    - Third-Party Integrations:
        - Test integrations with social media platforms, email marketing services, and analytics tools.
        - Ensure data syncs correctly and actions (like sharing on social media) function as expected.
- Localization and Internationalization:
    - Language Support:
        - Test language switch functionality (if applicable) to ensure content displays correctly in different languages.
        - Verify currency conversion and international shipping options (if applicable).
- Regression Testing:
    - Continuous Integration/Continuous Deployment (CI/CD):
        - Implement automated tests to run after each deployment to catch regressions early.
        - Perform manual regression tests periodically to ensure new updates haven’t introduced issues.
- Error Handling and Recovery Testing:
    - Form Validation:
        - Submit forms with incorrect or missing information to verify error messages appear correctly.
        - Test error recovery scenarios (e.g., session timeouts, network interruptions).
- Usability Testing:

- User Feedback:
    - Gather feedback from real users through surveys or usability testing sessions.
    - Identify areas where users encounter confusion or difficulty navigat

4.5 Defect Reporting and Tracking:

Defect reporting and tracking are crucial aspects of ensuring the quality and reliability of a street style store website. Here’s a structured approach to managing defect reporting and tracking effectively:

- ✓ Defect Reporting Process:
    - Steps to Report Defects:
        - **Identify the Defect:** Clearly describe the issue encountered, including steps to reproduce it and any relevant screenshots or logs.
        - **Provide Environment Details:** Mention the browser, device, and operating system where the defect was observed.
        - **Assign Category/Component:** Specify the affected area of the website (e.g., homepage, product page, checkout process).
        - **Include Additional Information:** Note any relevant context, such as user actions or specific conditions leading to the defect.
    - Defect Report Template:
        - **Title/Summary:** Briefly describe the issue.
        - **Description:** Detailed steps to reproduce, expected behavior, and actual behavior observed.
        - **Severity/Priority:** Impact and urgency leve.
        - **Category/Component:** Area of the website affected.
        - **Environment:** Browser, device, OS details.
        - **Attachments:** Screenshots, logs, or other supporting materials.

- ✓ Defect Tracking System:
    - ❖ Using a Defect Tracking Tool:
        - **Choose a Tool:** Use a dedicated defect tracking tool (e.g., JIRA, Bugzilla, Trello) to manage defects systematically.
        - **Create Defect Entries:** Input all reported defects into the tracking tool with relevant details.
        - **Assign Ownership:** Assign defects to developers or QA testers responsible for resolution.
        - **Track Status:** Update the status of each defect (e.g., open, in progress, fixed, closed) as it moves through the resolution process.
        - **Link to Requirements:** Link defects to corresponding requirements or user stories for traceability.
- ✓ Defect Lifecycle Management:
    - ❖ Defect Workflow:
        - **Open:** Defect reported and awaiting triage.
        - **Assigned:** Assigned to a developer or tester for investigation.
        - **In Progress:** Developer working on a fix.
        - **Resolved:** Fix implemented, pending verification.
        - **Closed:** Verified and confirmed as resolved.
    - ❖ Communication and Collaboration:
        - **Regular Updates:** Ensure stakeholders are informed of defect status changes.
        - **Collaboration:** Foster collaboration between developers, testers, and stakeholders to resolve defects efficiently.
        - **Feedback Loop:** Provide feedback on resolved defects to prevent recurrence.
- ✓ Metrics and Reporting:
    - ❖ Metrics to Track:
        - **Defect Density:** Number of defects per size of the project or per release.
        - **Defect Aging:** Time taken to resolve defects from open to close.

- **Defect Resolution Rate:** Percentage of defects resolved within defined timelines
- **Trend Analysis:** Identify recurring issues or patterns to improve processes.

Chapter 05

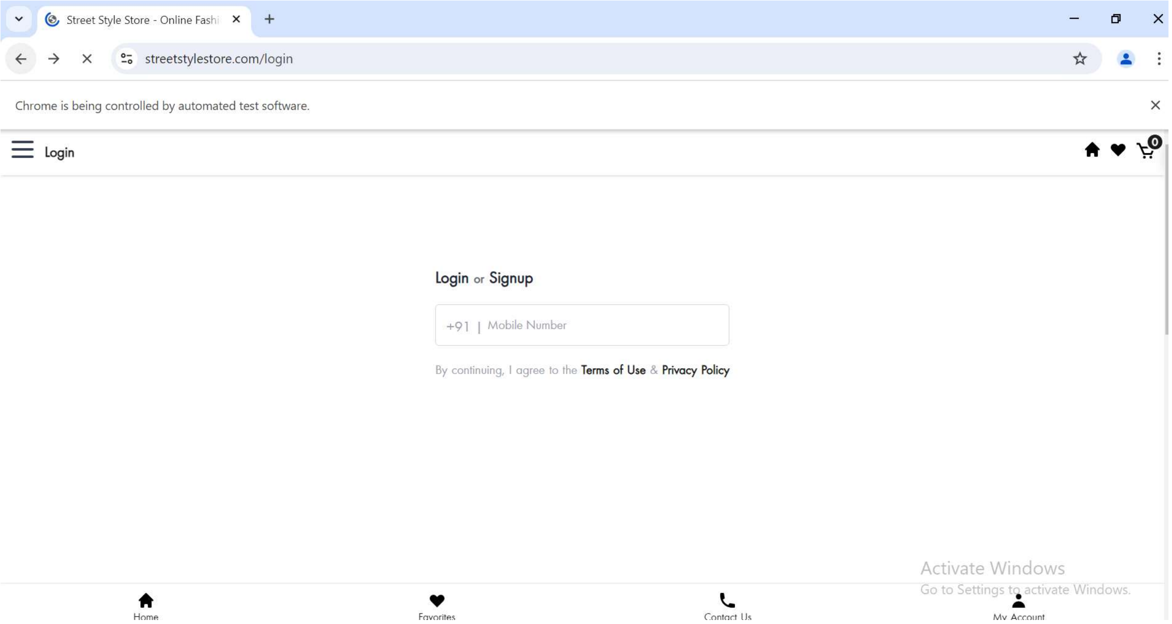
RESULT ANALYSIS

5.1 Test Cases:

Test Case ID	Description	Expected Output	Actual Output	Status
TC01	User registration with valid details	User is successfully registered	As expected	Pass
TC02	User registration with existing email	Registration fails with appropriate error message	User is registered with an existing email address	Fail
TC03	User login with valid credentials	User is successfully logged in	As expected	Pass
TC04	User login with invalid credentials	Login fails with appropriate error message	User successfully logged in with incorrectcredentials	Fail
TC05	Product search for existing product	Product page with search results is displayed	Product details page with options to add to cart	Pass
TC06	Add to cart for product availability	Product is successfully added to the cart	As expected	Pass
TC07	Product search fornon-existent product	No search results are displayed with appropriate message	Incorrect product displayed in search results	Fail

TC08	Add to cart - out of stock product	'Add to Cart' button is disabled or appropriate error shown	Product successfully added to cart despite being out of stock	Fail
TC09	Update the cart andremove product	Item is successfully removed from the cart	As expected	Pass
TC10	Update cart and change quantity	Cart reflects updated quantity and recalculates total price	Cart does not update after changing quantity	Fail
TC11	Checkout process forvalid payment	Order is successfully placed and confirmed	Payment gateway times out	Fail
TC12	User profile with update information	Profile information is updated successfully	Profile update confirmation message displayed	Pass
TC13	User profile andchange password	Password is successfully updated	Password change confirmation email received	Pass
TC14	Cross-Browser Compatibility	Login is successful across all tested browsers	Login fails in any browser due to rendering or compatibility issues	Fail
TC15	Mobile Responsiveness - Product Display	Product page is displayed correctly on mobile	As expected	Pass
TC16	Logout Functionality	User is successfully logged out and redirected to login page	As expected	Pass

5.2 Snapshots:



Chrome is being controlled by automated test software.

Login or Signup

+91 | Mobile Number

By continuing, I agree to the Terms of Use & Privacy Policy

Activate Windows
Go to Settings to activate Windows.

Fig 5.2.1:Log-In Page

The above fig5.2.1 Log-In Page to access your account and explore the latest street style trends, simply log in using your phone number.

Fig 5.2.1 : Home Page

e

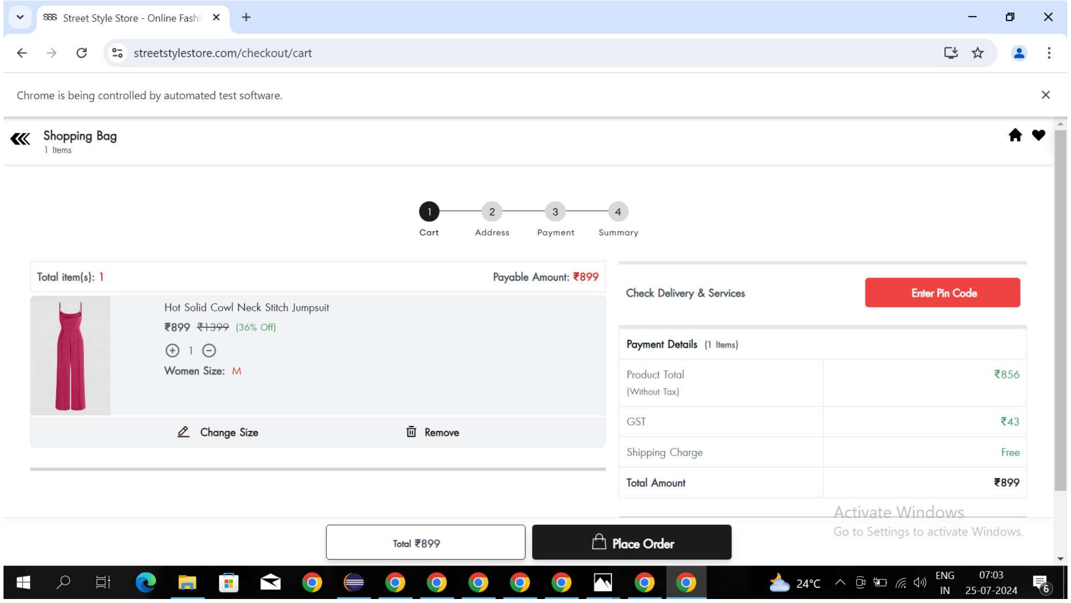
u

E

c

Fig 5.2.4 : Choosing Product

The above fig5.2.4 When choosing a product on the nopCommerce demo website, ensure that the selection process is smooth and accurate. Verify that product details, such as descriptions and prices, are displayed correctly, and that users can add items to their cart without issues.


Street Style Store - Online Fashi
streetstylestore.com/checkout/cart
Chrome is being controlled by automated test software.
Shopping Bag
1 Items
1 Cart
2 Address
3 Payment
4 Summary
Check Delivery & Services
Enter Pin Code
Change Size
Remove
Total ₹899
Place Order
Activate Windows
Go to Settings to activate Windows.
24°C
ENG IN
07:03
25-07-2024

Total item(s): 1 Payable Amount: ₹899

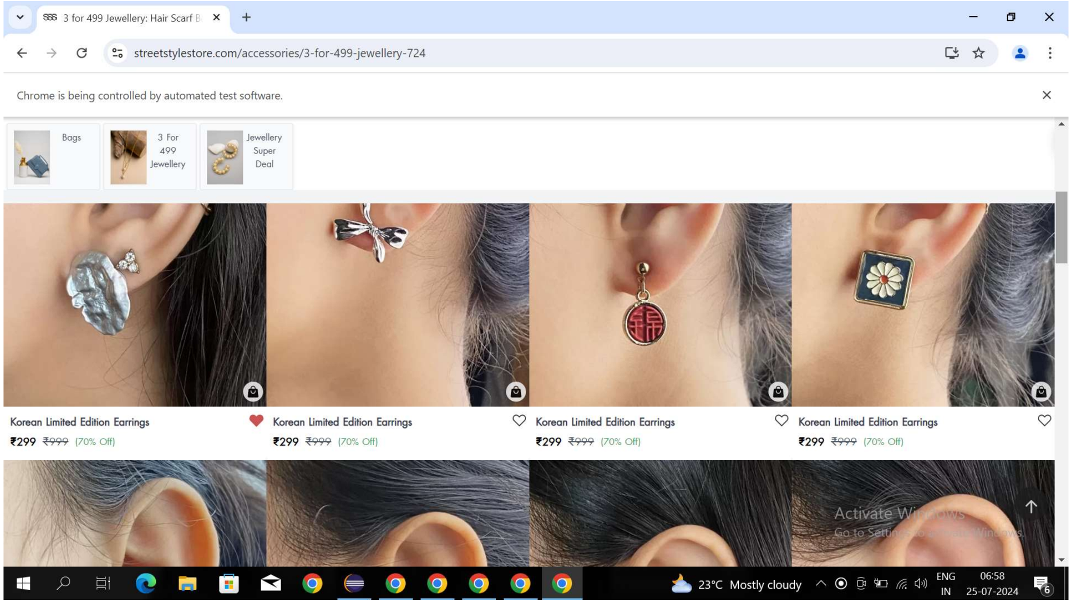
Hot Solid Cowl Neck Stitch Jumpsuit
₹899 ~~₹1399~~ (36% Off)
1
Women Size: M

Payment Details (1 Items)

Product Total (Without Tax)	₹856
GST	₹43
Shipping Charge	Free
Total Amount	₹899

Fig 5.2.7 : Add to Cart

The above fig 5.2.7 represents the product what we have select to order.


3 for 499 Jewellery: Hair Scarf B
streetstylestore.com/accessories/3-for-499-jewellery-724
Chrome is being controlled by automated test software.
Bags
3 For 499 Jewellery
Jewellery Super Deal
Activate Windows
Go to Settings to activate Windows.
23°C Mostly cloudy
ENG IN
06:58
25-07-2024

Korean Limited Edition Earrings
₹299 ~~₹999~~ (70% Off)

Korean Limited Edition Earrings
₹299 ~~₹999~~ (70% Off)

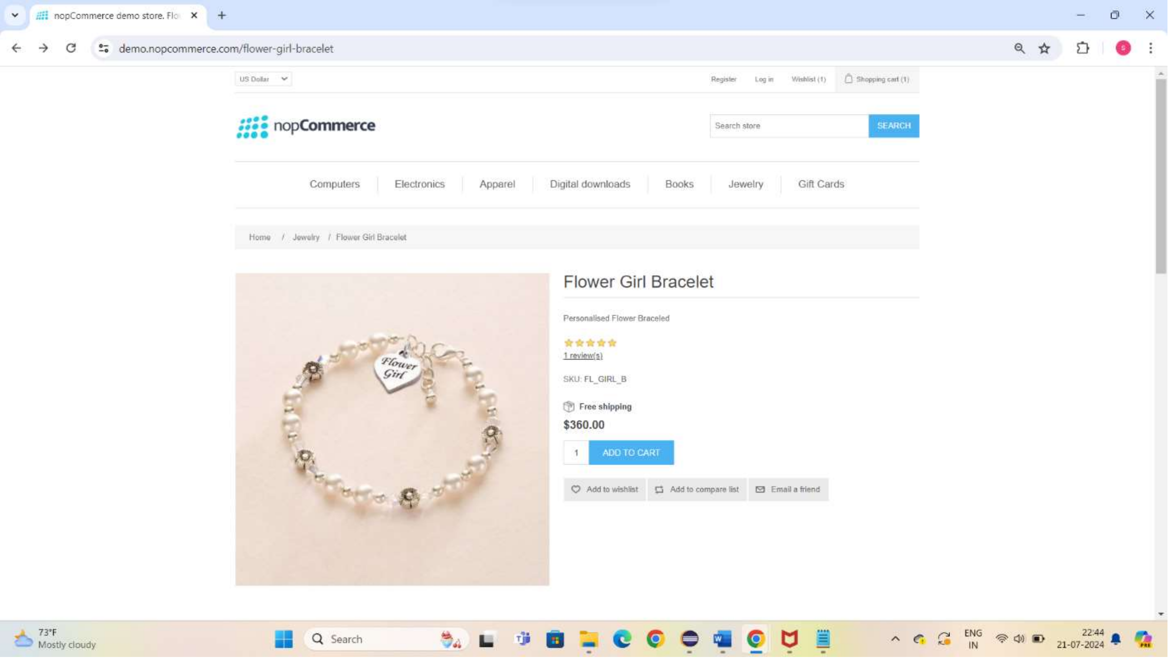
Korean Limited Edition Earrings
₹299 ~~₹999~~ (70% Off)

Korean Limited Edition Earrings
₹299 ~~₹999~~ (70% Off)

Fig 5.2.8 Added To Wishlist

The above fig 5.2.8 showing the product which has been added to wishlist based on the need.

The Fig 5.2.6 test case for displaying searched product details involves verifying that after performing a search, the product details such as name, price, description, and imageaccurately shown on the search results page.



Flower Girl Bracelet

Personalised Flower Braceled

1 review(s)

SKU: FL_GIRL_B

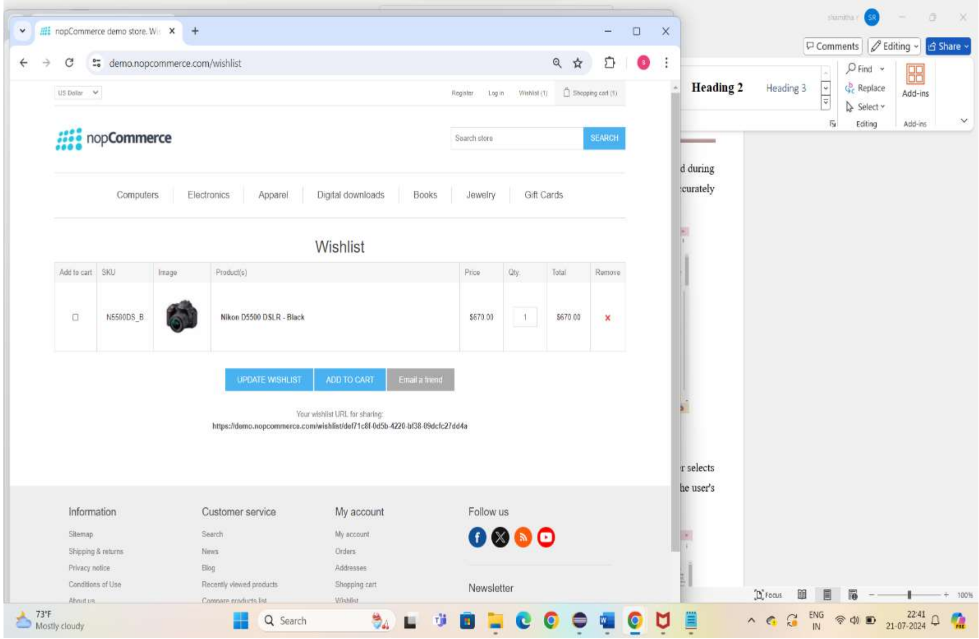
Free shipping

$360.00

1

Fig 5.2.7 Page displays the item Added to Cart

The fig 5.2.7 test case for adding an item to the cart verifies that, upon clicking the "Add to Cart" button on a product page, the cart updates correctly to reflect the addition, showing the correct item count and details.



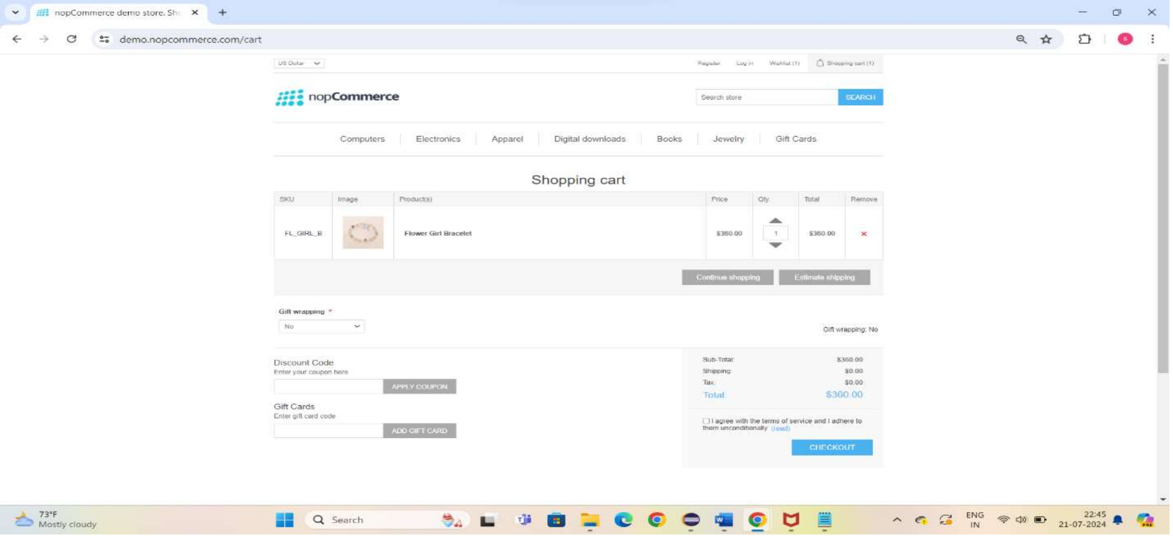
Wishlist

Add to cart	SKU	Image	Product(s)	Price	Qty.	Total	Remove
□	N5500DS_B		Nikon D5500 DSLR - Black	$670.00	1	$670.00	×

Your wishlist URL for sharing:
https://demo.nopcommerce.com/wishlist/def71c8f-0d5b-4220-bf38-09dcfc27dd4a

Fig 5.2.8 Page Display Item Added to Wishlist

The above fig5.2.8On the nopCommerce demo website, the wishlist feature should allow users to add products for future reference. Verify that items can be added and removed from the wishlist, and ensure that the wishlist is accurately updated and displayed.



Shopping cart

SKU	Image	Product(s)	Price	Qty.	Total	Remove
FL_GIRL_B		Flower Girl Bracelet	$360.00	1	$360.00	×

Gift wrapping * No

Gift wrapping: No

Discount Code
Enter your coupon here

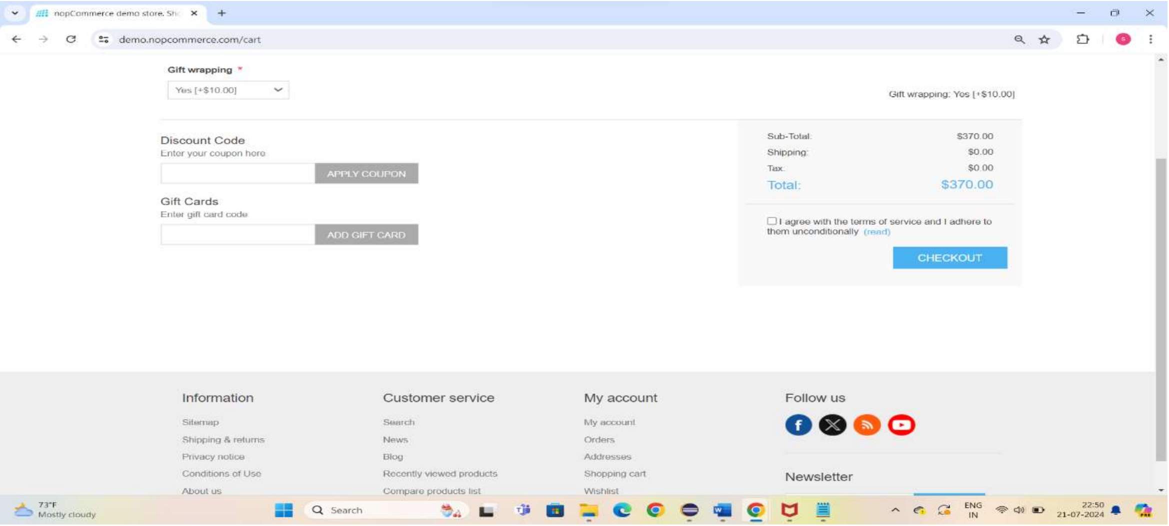
Gift Cards
Enter gift card code

Sub-Total:	$360.00
Shipping:	$0.00
Tax:	$0.00
Total:	$360.00

□ I agree with the terms of service and I adhere to them unconditionally (read)

Fig 5.2.9 Page Display Items in Shopping List

The above fig5.2.9 On the nopCommerce demo website, the shopping list should display all items added by the user with relevant details such as product names, prices, and quantities. Ensure that the list updates correctly when items are added or removed and that it reflects accurate information for each product.



demo.nopcommerce.com/cart

Gift wrapping *
Yes [+$10.00]

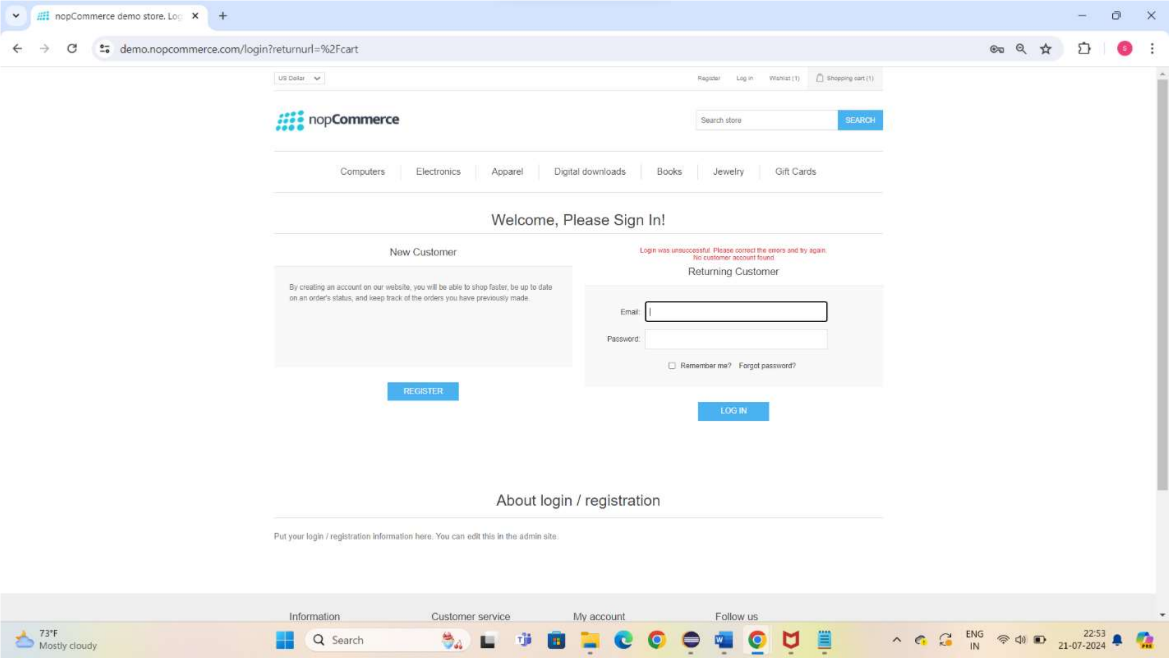
Gift wrapping: Yes [+$10.00]

Sub-Total:	$370.00
Shipping:	$0.00
Tax:	$0.00
Total:	$370.00

□ I agree with the terms of service and I adhere to them unconditionally (read)

Fig 5.2.10 Payment Method selection

The above fig5.2.8 When selecting a payment method on the nopCommerce demo website, ensure that all available payment options are displayed correctly. Verify that users can choose their preferred payment method, and that the selection process updates the checkout summary accurately.



Welcome, Please Sign In!

New Customer

By creating an account on our website, you will be able to shop faster, be up to date on an order's status, and keep track of the orders you have previously made.

Login was unsuccessful. Please correct the errors and try again.
No customer account found

Returning Customer

Email:

Password:

□ Remember me? Forgot password?

About login / registration

Put your login / registration information here. You can edit this in the admin site.

Fig 5.2.11 Test Case for Invalid

The above fig5.2.8 When testing invalid login scenarios on the nopCommerce demo website, ensure that the system appropriately handles incorrect credentials by displaying a clear error message. Verify that the user remains on the login page and that no unauthorized access is granted.

Chapter 06

CONCLUSION AND FUTURE WORK

6.1 Conclusion:

Implementing automated testing on our website has proven to be effective, significantly enhancing both the efficiency and accuracy of our quality assurance processes. Automated tests successfully validated essential features such as product selection, cart management, wishlist operations, and payment processing, ensuring that these functionalities worked consistently and met user expectations.

The automation of repetitive test cases reduced the time and effort required, while also increasing the reliability of the results by minimizing human error. Additionally, it improved user experience by quickly identifying and addressing issues related to login failures, empty cart scenarios, and other user interactions. The scalability of our automated testing approach supports ongoing development and adaptation, ensuring that our website remains robust and responsive to future changes. Overall, the use of automated testing has been a valuable asset, contributing to the overall quality and performance of our website.

6.2 Future Work:

Looking ahead, there are several key areas for enhancing Selenium Java testing for the nopCommerce demo website. Expanding test coverage is a priority, with plans to include additional features and scenarios such as promotional codes, customer reviews, and account management. Implementing the Page Object Model will improve the maintainability and readability of test scripts by organizing code more effectively. Optimizing test execution through parallel testing and integrating tests into a CI/CD pipeline will accelerate feedback and streamline the development process.

Enhancing data-driven testing by using external data sources will ensure thorough validation across various input scenarios. Improved test reporting with advanced frameworks like Allure or ExtentReports will provide more detailed insights into test results. Additionally, incorporating user interaction testing, cross-browser testing, and visual regression testing will ensure the website performs well across different browsers and maintains a consistent user experience. Finally, improving error handling and recovery mechanisms will help manage unexpected issues more gracefully. These future enhancements will significantly bolster the robustness, efficiency, and reliability of our automated testing efforts.

Chapter 07

REFERENCES

7.1 Textbooks:

1. Test Automation using Selenium Web Driver with Java by Mr. Navneesh Garg
2. JavaScript Selenium Handbook by Mr. Prithvi Varma Nallaparaju
3. Selenium with GITHUB by Anand Hooda
4. Learning Selenium Testing Tools by Raghavendra Prasad
5. Learn to code HTML and CSS by shay Howe.

7.2 Links:

- HTML Tutorial -https://www.w3schools.com/html
- CSS Tutorial -https://www.w3schools.com/Css
- JS Tutorial -https://www.w3schools.com/Js
- Selenium Tutorial - https://www.javatpoint.com/selenium-tutorial
- Selenium WebDriver Tutorial - https://www.javatpoint.com/selenium-webdriver