Micro Ecommerce Project

Table of Contents

[1 Introduction 3](#_Toc515524393)

[1.1 Purpose 3](#_Toc515524394)

[1.2 Project Overview 3](#_Toc515524395)

[2 Scope 3](#_Toc515524396)

[2.1 In-Scope 3](#_Toc515524397)

[2.2 Out-of-Scope 3](#_Toc515524398)

[3 Testing Strategy 3](#_Toc515524399)

[3.1 Test Objectives 3](#_Toc515524400)

[3.2 Test Assumptions 3](#_Toc515524401)

[3.3 Data Approach 3](#_Toc515524402)

[3.4 Level of Testing 3](#_Toc515524403)

[3.5 Smoke Testing 4](#_Toc515524404)

[3.6 Functional Testing 4](#_Toc515524405)

3.8 Integration Testing………………………………………………………………………………………………………………

3.9 System Testing……………………………………………………………………………………………………………………

[3.10 User Acceptance Testing 4](#_Toc515524406)

[3.11 Regression Testing 5](#_Toc515524407)

[4 Execution Strategy 5](#_Toc515524408)

[4.1 Entry Criteria 5](#_Toc515524409)

[4.2 Exit criteria 6](#_Toc515524410)

[4.3 Validation and Defect Management 6](#_Toc515524411)

[5 Environment Requirements 7](#_Toc515524412)

[5.1 Test Environments 7](#_Toc515524413)

[6 Significantly Impacted Division/College/Department 7](#_Toc515524414)

[7 Dependencies 7](#_Toc515524415)

# Introduction

## Purpose

The purpose of this test plan is to outline the testing strategy and approach for ensuring the quality, reliability, and security of the e-commerce website. By conducting thorough testing across various dimensions, we aim to identify and rectify any issues or defects, thereby enhancing the overall user experience and promoting customer satisfaction. The purpose of this test plan is to outline the testing strategy and approach for ensuring the quality, reliability, and security of the e-commerce website. By conducting thorough testing across various dimensions, we aim to identify and rectify any issues or defects, thereby enhancing the overall user experience and promoting customer satisfaction.

**1.2. Project Overview**

The e-commerce website aims to provide a platform for users to browse, search, and purchase products online. Key features include user registration, product catalogue, shopping cart management, checkout process, payment integration, order tracking, and account management functionalities. The website targets a diverse audience and needs to be accessible, intuitive, and secure across different devices and platforms. The success of the project hinges on delivering a seamless and enjoyable shopping experience for users while maintaining the integrity and confidentiality of their personal and financial information.

**2.SCOPE:**

**2.1. IN-SCOPE:**

  1.User registration and login functionality.  
     2.Product browsing, searching, and filtering.  
     3.Adding products to the cart and managing the cart.  
     4.Checkout process including shipping address, payment methods, and order summary.  
     5.Order confirmation and email notifications.  
     6.Account management such as profile updates and password resets.  
     7.Product management from the admin side (if applicable).  
     8.Payment gateway integration and transaction processing.  
     9.Compatibility testing across different browsers and devices.  
     10.Security testing including SSL encryption, data encryption, and secure payment processing.

# Testing Strategy

The Ecommerce Testing Strategy determine the project approach to testing. The strategy looks at the characteristics of the system to be built the project timeline and budget and plans the breadth and depth effort the Testing Strategy will influence tasks related to test planning, Test types, test scripts development and test execution.

It's more important than ever to make sure your ecommerce website is functioning correctly and is as user friendly as possible. Recent ecommerce trends have shown an increased presence of online stores on the internet therefore, creating more competition. One of the best ways to make sure your ecommerce website is standing out above your competitors is to have a solid test plan in place.

## Test Objectives

**Objective Identification:**

Identify the objectives of the testing effort, such as validating data collection, assessing alerting mechanisms, evaluating performance, and ensuring security.

Ensure that the test objectives align with the goals of the project and the expectations of stakeholders.

**Testing Methods and Techniques:**

Determine the testing methods and techniques to be used, considering factors such as the complexity of the system, available resources, and project timeline.

Consider a combination of manual and automated testing approaches to achieve comprehensive coverage and efficiency.

Define specific methodologies for functional testing, performance testing, security testing, and integration testing.

**Test Environment Setup:**

Establish the test environment, including the hardware, software, and network infrastructure required for testing.

Configure test environments to mirror production environments as closely as possible, including network topologies, device configurations, and traffic patterns.

Ensure that the test environment is isolated from production systems to avoid interference with live operations.

## Test Assumptions

In software testing, risks are the possible problems that might endanger the objectives of the project stakeholders. It is the possibility of a negative or undesirable outcome. A risk is something that has not happened yet and it may never happen; it is a potential problem.

You can implement risk-based testing when:

There are resource, budget, and time constraints for the project.

You are performing security testing in cloud computing environments.

You need to conduct testing for projects with high-risk factors.

There is an ongoing process where the project is continuously changing.

## Data Approach

Prepare test data sets representing various network conditions, events, and anomalies.

Include both synthetic data and real-world data captured from production environments to simulate realistic scenarios.

Ensure that test data adequately covers all functionalities and features of the network monitoring system.

## Level of Testing

**Test Case Development:**

Develop detailed test cases for each test objective identified earlier.

Ensure that test cases are comprehensive, covering various scenarios, edge cases, and performance benchmarks.

Include both positive and negative test cases to validate expected behaviour and handle exceptions.

|  |  |  |
| --- | --- | --- |
| **Test Type** | **Description** | **Responsible Parties** |
| **Smoke Testing** | It is used to Determine if a new software build is ready for the next testing phase Smoke tests verify whether the most important features work as expected and that there are no showstopper issues in the build that can potentially lead to blocking the entire testing team. It helps in deciding if the build is flawed or not and hence, prevents the entire team from wasting time or resources.  (Admin Registration and login page also User registration and login page add to cart, home page view products details seller login page, registration page home page adds the project and payment) where we will test on above features and verify whether the website is stable for the next level of testing. | **Bhavana, Bhavya** |
| **Functional Testing** | Functional Testing is a type of software testing that validates the software system against the functional requirements/specifications. The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements.  [**Unit testing**](https://www.geeksforgeeks.org/unit-testing-software-testing/) is the type of functional testing technique where the individual units or modules of the application are tested. It ensures that each module is working correctly.  In [**Integration testing**](https://www.geeksforgeeks.org/software-engineering-integration-testing/)**,** combined individual units are tested as a group and expose the faults in the interaction between the integrated units.  [**System testing**](https://www.geeksforgeeks.org/system-testing/) is a type of software testing that is performed on the complete integrated system to evaluate the compliance of the system with the corresponding requirements.  [**Usability testing**](https://www.geeksforgeeks.org/usability-testing/) is done to measure how easy and user-friendly a software application is.  [**User acceptance testing**](https://www.geeksforgeeks.org/acceptance-testing-software-testing/) is done by the client to certify that the system meets the requirements and works as intended. It is the final phase of testing before the product release.  [**Regression testing**](https://www.geeksforgeeks.org/software-engineering-regression-testing/)is done to make sure that the code changes should not affect the existing functionality and the features of the application. It concentrates on whether all parts are working or not. |  |
| **Non-Functional Testing** | Non-Functional testing is essential for confirming the software’s reliability and functionality. The [Software Requirements Specification (SRS)](https://www.browserstack.com/guide/software-requirement-specifications-in-agile) serves as the basis for this software testing method, which enables quality assurance teams to check if the system complies with user requirements. Increasing the product’s [usability](https://www.browserstack.com/guide/what-is-usability-testing), effectiveness, [maintainability](https://www.browserstack.com/guide/maintainability-testing), and portability is the goal of non-functional testing. It aids in lowering the manufacturing risk associated with the product’s non-functional components. ****Performance Testing**** [Performance testing](https://www.browserstack.com/guide/performance-testing) eliminates the causes of the software’s sluggish and constrained performance. The software’s reading speed should be as quick as possible. One must create a well-organized and precise specification about the desired speed for Performance Testing. Otherwise, it won’t be evident if the test is a success or a failure. Example: When 1000 users use an application simultaneously, the load time shouldn’t exceed 5 seconds.  **Tools Used:** JMeter |  |
|  |  |  |

## SMOKE TESTING (Manual Testing)

To Check whether the website is stable for testing Checking the main functionality of the website like

* Admin Login registration page able to login to the home page customer care support Admin dashboard logout.
* Seller Login Registration Page able to login to the home page able to add the product see the order status and payment logout.

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Bhavana Bhavya | (Admin) | QA |
| Bhavana Bhavya | (Seller) | QA |
| Bhavana Bhavya | (Buyer) | QA n |

* Buyer should be able to login and register to website search the product view the details of the product add to cart manage the cart procced to dispatch by doing the payment order confirmation

## FUNCTIONALY TESTING (Functional testing using Selenium)

Checking whether each feature’s functionality is working according to the customer requirements.

**Admin**

Register: need to verify name filed , email address, password, and optional fields for additional details. And submit button whether it is working according to user requirement.

Login: Need to verify the username field and password field working and login button and forget password button working according to the user needs.

Admin Dashboard:

1.All selling products details

2.All buyer's details

3.All Seller's details

4.can see whole production amount from sellers in ist

5.Most selling Product

6.Shipping/ Tracking

7.Reviews or customer care messages

8.Logout

**Seller**

Register

Login

Customer Support

Sellers Dashboard: Add, Delete, Update, Number of stocks, Offer

0rders List: 2 status -> PENDING, ORDER SENT

1.ITEMS

3.PAYMENTS: total revenue

4.Profile - Email, phone number, Seller ld.,

5.Logout

**Buyer**

Register

Login

home page Customer support

search bar catalog

add to cart, rating

Buyers Dashboard:

1.Profile

2.Orders History

3.Order tracker

4Cart

5.Logout

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Bhavana Bhavya | (Admin) | QA |
| Bhavana Bhavya | (Seller) | QA |
| Bhavana Bhavya | (Buyer) | QA |

**3.5 Integration Testing**

Integration testing is known as the second level of the software testing process, following unit testing. the interface between two software units or modules It focuses on determining the correctness of the interface The purpose of integration testing is to expose faults in the interaction between integrated units. Once all the modules have been unit-tested, integration testing is performed.

1. Check the integrity between admin and the seller:

Admin will check the seller products details, Quality &Quantity details from seller upload their products in site.

2.Check the integrity between buyer and the seller

Buyer will buy the products from sellers

3.Check the integrity between buyer and the admin

Admin will check all the product details, payment details and shipping details from buyers side

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Bhavana Bhavya | (Admin) | QA |
| Bhavana Bhavya | (Seller) | QA |
| Bhavana Bhavya | (Buyer) | QA |

## 3.6 System Testing

software testing that is performed on the complete integrated system to evaluate the compliance of the system with the corresponding requirements.

Testing Complete end to end flow from the end user point perceptive

Admin will the check the products details like quantity, quality, type of product from the seller side then buyers will search for a product and see the ratings, reviews of the product then add to the cart and select the quantity of product then do the payment process then the order will conform by the seller side and seller will send the tracking id to buyers like when the product will deliver.

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Bhavana Bhavya | (Admin) | QA |
| Bhavana Bhavya | (Seller) | QA |
| Bhavana Bhavya | (Buyer) | QA |

**3.7 Acceptance Testing**

The major aim of this test is to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not.

Acceptance testing is done by the end user perspective by the customers.

**3.8 Regression Testing**

Regression testing is a type of software testing conducted after a code update to ensure that the update introduced no new bugs. This is because new code may bring in new logic that conflicts with the existing code, leading to defects. Usually, QA teams have a series of regression test cases for important features that they will re-execute each time these code changes occur to save time and maximize test efficiency.

To Test The important features like add to cart, payment, add products, order status, admin site, login, registration

# **4.Execution Strategy**

# Execute the test cases according to the defined test strategy and test plan.

Utilize appropriate tools and techniques for executing manual and automated tests.

Monitor test execution progress, document test results, and capture relevant metrics and performance data.

## 4.1 Entry Criteria

1.The Tester should determine all the elements to be tested on the website is called as requirements

2. Test Plan document should be prepared.

3. we need to design the test case or test script for automation

4.Test environment and test data should be present

5. we need to do smoke testing before starting any testing so that we will get the conformation the software is stable.

|  |  |  |  |
| --- | --- | --- | --- |
| **Entry Criteria** | **Test Team** | **Technical Team** | **Notes** |
| *Test environment(s) is available* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png |  |  |
| *Test data is available* |  |  |  |
| *Code has been merged successfully* |  |  |  |
| *Development has completed unit testing* |  |  |  |
| *Test scripts are completed, reviewed and approved by the Project Team* |  |  |  |

## Exit criteria

|  |  |  |  |
| --- | --- | --- | --- |
| **Exit Criteria** | **Test Team** | **Technical Team** | **Notes** |
| *100% Test Scripts executed* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png |  |  |
| *90% pass rate of Test Scripts* |  |  |  |
| *No open Critical and High severity defects* |  |  |  |
| *All remaining defects are either cancelled or documented as Change Requests for a future release* |  |  |  |
| *All expected and actual results are captured and documented with the test script* |  |  |  |
| *All test metrics collected based on reports from daily and Weekly Status reports* |  |  |  |
| *All defects logged in Defect Tracker/Spreadsheet* |  |  |  |
| *Test environment cleanup completed and a new back up of the environment* |  |  |  |

## Validation and Defect Management

Defects will be logged using a designated tracking tool (e.g., Jira).

Defect severity levels: Critical, Major, Minor.

Defect priorities: High, Medium, Low.

Defects found during the Testing should be categorized as below:

|  |  |
| --- | --- |
| **Severity** | **Impact** |
| *1 (Critical)* | * *Functionality is blocked and no testing can proceed* * *Application/program/feature is unusable in the current state* |
| *2 (High)* | * *Functionality is not usable and there is no workaround but testing can proceed* |
| *3 (Medium)* | * *Functionality issues but there is workaround for achieving the desired functionality* |
| *4 (Low)* | * *Unclear error message or cosmetic error which has minimum impact on product use.* |

**5 Environment Requirements**

5.1 TEST ENVIRONMENTS

Establish the test environment, including the hardware, software, and network infrastructure required for testing.

Configure test environments to mirror production environments as closely as possible, including network topologies, device configurations, and traffic patterns.

Ensure that the test environment is isolated from production systems to avoid interference with live operations.

Automation testing would comprise of:

- Functional testing using Selenium

- Performance testing would be conducted using JMeter

- API Testing using Postman/SOAPUI