**Test Plan**

Table of Contents

[1 Introduction: 2](#_Toc163943109)

[1.1 Purpose 2](#_Toc163943110)

[1.2 Project Overview 2](#_Toc163943111)

[2 Scope: 2](#_Toc163943112)

[2.1 In-Scope 2](#_Toc163943113)

[2.2 Out-of-Scope 2](#_Toc163943114)

[3 Test Approach: 2](#_Toc163943115)

[3.1 Test Objectives 2](#_Toc163943116)

[3.2 Tools and Technologies 3](#_Toc163943117)

[3.3 Level of Testing 3](#_Toc163943118)

[4 Test Environment 3](#_Toc163943119)

[5 Test Cases 4](#_Toc163943120)

[6 Test Data 5](#_Toc163943121)

[6.1 Sample Test Data: 5](#_Toc163943122)

[6.2 Steps to Generate Test Data: 5](#_Toc163943123)

[6.2.1 User Credentials: 5](#_Toc163943124)

[6.2.2 Additional Test Data: 5](#_Toc163943125)

[6.3 Test Data Management: 5](#_Toc163943126)

[7 Execution Strategy 6](#_Toc163943127)

[7.1 Entry Criteria 6](#_Toc163943128)

[7.2 Exit criteria 6](#_Toc163943129)

[8 Risks and Mitigation 7](#_Toc163943130)

[8.1 Potential Risks 7](#_Toc163943131)

[8.2 Risk and Defect Management 7](#_Toc163943132)

[8.3 Mitigation Strategies: 8](#_Toc163943133)

[9 Reporting and Documentation 8](#_Toc163943134)

[10 Conclusion 8](#_Toc163943135)

# Introduction:

## Purpose

The test plan is dedicated to ensuring the quality and reliability of the Sign-in, Sign-up, and Authentication modules within the Whizlabs Retail Platform. This involves rigorous testing to verify that these modules perform according to specifications, meet predefined standards, and deliver a seamless user experience.

## Project Overview

Whizlabs' E-learning platform is a cornerstone for learners in various technological domains. The modules under scrutiny encompass critical user authentication processes. The plan includes comprehensive testing methodologies to validate the functionality and security of the authentication processes. This encompasses functional verification, usability evaluation, performance testing, and security validation to identify and mitigate potential vulnerabilities, ensuring the protection of user data and the integrity of the platform.

# Scope:

## In-Scope

- Sign-in module.

- Sign-up module.

- Authentication processes.

## Out-of-Scope

- Deep integration testing with third-party authentication providers

- Advanced security testing beyond standard authentication procedures, such as penetration testing

# Test Approach

## Test Objectives

The objective of the test approach is to establish a structured methodology for evaluating the functionality, usability and performance of the Sign-in, Sign-up, and Authentication modules.

By defining clear testing methodologies, techniques, tools, and environments, the test approach facilitates efficient and effective testing while aligning with project goals and objectives. It aims to ensure reliability, meet requirements, and provide a secure user experience.

Top of Form

## Tools and Technologies

- Java Selenium for automated functional testing

- Apache J Meter for performance testing

## Level of Testing

List the types of testing to be performed.

|  |  |
| --- | --- |
| **Test Type** | **Description** |
| **Functional testing** | Ensures that the software features work correctly according to specified requirements. |
| **Smoke testing** | Rapidly verifies critical functionalities post-deployment, determining if further testing is viable. |
| **Integration Testing** | Validates interactions between system components to ensure seamless integration and functionality. |
| **System Testing** | Evaluates end-to-end functionality to confirm compliance with requirements and user expectations. |
| **Ad-hoc testing** | Informally explores software to uncover defects and usability issues intuitively. |
| **Usability testing** | Evaluates how easy and intuitive the software is to use, focusing on user experience. |
| **Performance testing** | Checks how well the software performs under different conditions, measuring speed, scalability, and stability. |

# Test Environment

* **Operating System**:

- Windows 11 Pro

* **Hardware:**

- Processor: Intel Core i7-10700K

- RAM: 16GB DDR4

- Storage: 512GB SSD

* **Software:**

- Application Server: Apache Tomcat 9.0

- Database Server: MySQL 8.0

- Web Browser: Google Chrome, Mozilla Firefox

# Test Cases

Please refer to the Test Cases Template in here [Test Case Template](Templates/Test%20Case%20Template.xlsx)

# Test Data

## Sample Test Data:

- Name: test user

- Email:sriramsri2711@gmail.com

- Password: Test@123

## Steps to Generate Test Data:

### User Credentials:

1. Create a new user account in the system.

2. Use the provided Name, Email and Password during the registration process.

3. Verify that the user account is successfully created and can be used for sign-in.

### Additional Test Data:

- For testing different scenarios, create multiple user accounts with varying characteristics (e.g., different lengths of usernames and passwords, special characters in passwords).

- Generate test data to simulate scenarios such as account lockouts after multiple failed login attempts or password expiration.

## Test Data Management:

- Data Privacy: Ensure that sensitive information such as passwords is stored securely and handled in compliance with privacy regulations.

- Data Reusability: Maintain a repository of test data for reuse in regression testing and future test cycles.

- Data Variation: Generate diverse test data to cover various test scenarios and edge cases.

- Data Cleansing: Regularly update and cleanse test data to reflect changes in system requirements or user profiles.

# Execution Strategy

## Entry Criteria

* The entry criteria refer to the desirable conditions in order to start test execution
* Entry criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 case are completed, reviewed and approved by the Test lead. |  |  |  |

## Exit criteria

* The exit criteria are the desirable conditions that need to be met in order proceed with the implementation.
* Exit criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 Cases |  |  |  |
| 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 |  |  |  |

# Risks and Mitigation

## Potential Risks

- Performance bottlenecks under high load

- Security vulnerabilities in authentication processes, such as brute force attacks or insecure session management

## Risk and Defect Management

* Risks will be assessed based on likelihood and impact, utilizing risk matrices to prioritize mitigation efforts.
* How defect should be managed
  + It is expected that the testers execute all the scripts in each of the cycles described above.
  + The defects will be tracked through Defect Tracker or Spreadsheet.
  + It is the responsibility of the tester to open the defects, retest and close the defect.

Defects found during the Testing should be categorized as below:

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

## Mitigation Strategies:

- Performance testing to identify and address scalability issues, optimizing system architecture and resource allocation.

- Regular security assessments, including vulnerability scanning and code reviews, to identify and mitigate security risks proactively.

# Reporting and Documentation

* Comprehensive test reports will be generated to document test activities, findings, and recommendations.

Please refer to the Defect report template in here [Defect report template](Templates/Defect%20report%20template.xlsx)

Please refer to the Test Case template in here [Test Case Template](Templates/Test%20Case%20Template.xlsx)

* Reports will include detailed descriptions of test cases, test results, defects identified, and recommendations for improvement.
* Regular stakeholder communication and reporting will ensure transparency, facilitate decision-making, and promote alignment with project objectives.

# Conclusion

Action items and follow-up tasks will be identified based on the analysis of test results, addressing any identified issues or gaps in the Sign-in, Sign-up, and Authentication modules.