

Master Test Plan

Sample Template

Date

06/10/2022

Document Approval

The approval of this document indicates that the signer has read, understood, and approves the content of the Test Plan. If the person disagrees with or has found major discrepancies in the document as written, they should return the unsigned document and a description of the issues to the author. If the discrepancy is minor, they return the document along with the conditions of their approval to the author.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Role | Name | Title | Signature | Date |
| Author | Pavithran | QA Engineer 1 | Pavithran | 06/10/2022 |

Document Revision History

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Author | | Change | Version | | Date | | |
| Gautam Mahapatra | Initial document | | | 0.1 | |  |

\* Minor versions (1.1, 1.2, 1.3 etc.) should be recorded here while the document content is being written and agreed

\* Approved / Published versions should only contain a summary of the changes included in each published or major version (1.0, 2.0 etc) i.e. the details relating to minor versions should be removed when the new version is approved

Table of Contents

[**1. Purpose**](#_gjdgxs) **4**

[1.1 Project Overview](#_30j0zll) 4

[**2. Scope**](#_dcouczmdt629) **5**

[2.1 Feature Testing](#_t61ktdncyw6l) 6

[2.2 Compatibility testing](#_9h40refrm67c) 7

[2.3 Integration testing](#_686799xwef01) 7

[**3. Definitions**](#_3dy6vkm) **8**

[**4. Responsibilities**](#_4d34og8) **9**

[**5. Test Approach**](#_2s8eyo1) **10**

[5.1 Sample Application Unit (Development) Testing](#_17dp8vu) 10

[5.1.1 Entry and Exit Criteria](#_sqmvomuhluie) 10

[5.2 Sample Application QA Testing](#_35nkun2) 10

[5.2.1 Entry and Exit Criteria](#_4hn9gb48fpr1) 11

[5.3 User Acceptance Testing](#_1ksv4uv) 12

[5.3.1 Entry and Exit Criteria](#_2nx4wbwc5s6w) 12

[**6. Risk Management**](#_44sinio) **13**

[6.1 User Stories](#_p2s31entxx0o) 14

[6.2 Assumptions and Dependencies](#_2jxsxqh) 14

[**7. Testing Phases / Milestones / Resources**](#_6czvglqrsi7h) **15**

[**8. Test Environments**](#_1y810tw) **16**

[**9. Defect Management**](#_4i7ojhp) **17**

[**10. Test Metrics**](#_1ci93xb) **19**

[10.1 User Stories/Defects to be Tested](#_qq8z1m4jqq1r) 19

[10.2 Test Cases](#_xdnc855wzt21) 20

[**11. References**](#_cdzcw5jywcf8) **21**

# 1. Purpose

This master test plan defines the testing strategy and methodology for the **Sample Template Project**. It describes the testing strategy and approach that QA will use to validate the expected level of quality prior to release. It is the vehicle used to strategize, control, manage, and communicate the overall test approach, while clarifying the major tasks and challenges associated with the test effort.

## 1.1 Project Overview

1. As part of this project, Sample Application intends to develop a new website for **Sample Template Project** that will help visitors consume site content efficiently.
2. This new website will also have a modern eye catching UI with host of new features.
3. The underlying CMS system will allow admins to manage website content.

# 2. Scope

Below table outlines various testing activities which are in scope of testing for **Sample Template Project** Website Design project.

|  |  |  |
| --- | --- | --- |
| Method | Scope | Notes |
| Server-Side API Testing | In scope | Testing the server-side APIs using Postman. |
| Client-Side UI Testing | In scope | 1. Testing the website in different browsers.  2. Testing admin portal. |
| Performance Testing | In scope | Performance testing will be carried out once the application is stable |
| Security Testing (Penetration & Dynamic) | Out of scope |  |
| Smoke Testing | In scope | 1. QA will be working on creating a smoke Test suite for API & functional Tests.  2. QA would be running these tests whenever a new build is deployed. |
| User Acceptance Testing (UAT) | In scope | **Sample Template Project** will take care of the UAT. They will document and forward defects which will be fixed by Sample Application. |
| Regression Testing | In scope | Regression suite would entail the P1 & P2 test cases from the general test suite. |
| Automation Testing | In scope | Automation Testing will be carried out once application is 80% stable. |
| End to End Functional Testing | In scope | 1. QA will be working on creating functional test cases for all the features.  2. QA would be running these tests and provide test report whenever new feature is developed and ready to test. |

## 2.1 Feature Testing

The following table describes, at a high level, the features, which will be in scope as part of testing efforts.

|  |  |
| --- | --- |
| Features | Notes |
| **Student Role:**  1. Sign up  2. Login  3. Course Enrollment  4. Course Consumption  5. Certificate |  |
| **Super Admin:**  1. Full Access for Website  2. Create/Edit User and User Groups Privileges  3. Dashboard Access  4. Payment Management |  |
| **Admin:**  1. Switch Role (if other role permission is given)  2. Course Creation  3. Create/Edit User and User Groups  4. Membership/Subscription Management  5. Organization Management  6. Instructor and Student management  7. Event Management  8. Payment Management |  |
| **Organization:**  1. Switch Role (if other role permission is given)  2. Department and Team Roles Management  3. Course Creation  4. Assign Department, Team and User for the organization |  |
| **Department:**  1. Switch Role (if other role permission is given)  2. Team and Individual role management  3. Able to Assign Team to Department and User to Team.  4. Can Submit Course to Organization for Publishing  5. Can Submit Event to Organization for Hosting |  |
| **Team:**  1. Switch Role (if other role permission is given)  2. Individual Role management  3. Able to Assign User (Individual) to team  4. Can Submit created Event and Course to Department for Review and Submission |  |
| **Instructor:**  1. Switch Role (if other role permission is given)  2.Create/Edit course |  |
| **Speaker:**  1. It is not a Role  2. Just a user who is going to Speak in an live Event or Live course session. |  |
|  |  |

## 2.2 Compatibility testing

The website will be tested in below operating system - browser combinations.

|  |  |  |
| --- | --- | --- |
| Operating system/Platform | Browser with version # | Notes |
| Microsoft Windows | Google Chrome latest |  |
| Apple Mac | Safari latest |  |

## 2.3 Integration testing

The following table describes the integration of the website with any 3rd party services.

|  |  |  |
| --- | --- | --- |
| System/Integration | Scope | Notes |
| Facebook integration | Out of scope |  |
| LinkedIn Integration | Out of scope |  |
| Gmail Integration | In scope |  |

3. Responsibilities

|  |  |  |
| --- | --- | --- |
| Role | Responsibility | Organization |
| Project Manager | * Generation of acceptance criteria based on user stories * Ensuring that Quality Assurance activities are planned and executed as required * Liaise between third parties and QA for access to third party systems and applications | Sample Application |
| Technical Lead | * Creation of user stories prior to starting a new Sprint * Documentation of requirements | Sample Application |
| Quality Assurance | * Author the Master Test Plan * Test case creation and execution * Daily/Weekly status reporting. * Provide support in review of test cases and defect triage during UAT phase | Sample Application |
| Customer or user representative | * Execute User Acceptance Testing * Review and Approve documents related to UAT | **Sample Template Project** |

# 

# 4. Test Approach

This section describes the approach taken to each separate phase of testing. The nature and length of these phases may vary depending on project plan.

## 4.1 Sample Application QA Testing

**Ownership – Sample Application QA team**

**Smoke testing:** Smoke test will be performed on each build delivered to QA. Acceptance of build for further functional testing will be subject to successful completion of smoke test with no blockers. Smoke will be limited to verification of high priority test cases from test suite.

**Functional testing:** Functional testing would entail testing website functionality on different browsers. Features of website will be tested with positive and negative scenarios to make sure that the website meets the functional specifications.

**Integration testing:** This will involve testing of website with any 3rd party APIs.

**Regression testing**: Regression testing will be done before each sprint to ensure the newly added features or bug fixes does not break the existing website functionality. This will be done by preparing a set of high priority test cases and executing them.

**Compatibility testing**: This will ensure the website functionality is consistent across different platforms/OS-browser combinations.

### 4.1.1 Entry and Exit Criteria

|  |  |  |  |
| --- | --- | --- | --- |
| Test Phase | Entry Criteria | Exit Criteria | Documentation |
| Smoke testing | * Build has been successfully deployed. * Release notes are available. | * Smoke test suite has passed successfully. | * Sanity test results |
| Functional Testing | * Successful completion of QA sanity * Availability of Release Notes * Signed off stories and acceptance criteria * Master test plan and test scripts are ready, reviewed and approved. | * Test Coverage of 100% of the stories. * Test Execution of 100 % of the test scripts in qTest. * Make sure all the APIs return expected response * Completion of all QA tasks and subtasks in JIRA * No open Blocker or Showstopper | * Test Execution Report * Test Coverage Report * Test Metrics as described in section * Signoff report for each Sprint |
| System Integration Testing | * Functional test has passed successfully. * Access to third party systems and application is available to QA team * Test data required to execute the integration test cases is available. | * All system integration test cases have been executed. * There are no Blocker or Showstoppers | * Test Execution Report |

## 5.3 User Acceptance Testing

**Owned By – Sample Template Project Technologies**

User Acceptance testing (UAT) will be done by the **Sample Template Project** Technologies, to validate the developed website against the business rules and requirements mentioned in approved JIRA user stories.

**UAT Scope Binding Assumptions:**

* UAT plan would be prepared and owned by **Sample Template Project** Technologies. Sample Application QA team can help by reviewing the same.
* Any items deemed ‘out of scope’ based on the signed off SOW documents will be treated as change requests and if required, will be planned for subsequent releases

### 5.3.1 Entry and Exit Criteria

|  |  |  |  |
| --- | --- | --- | --- |
| Test Phase | Entry Criteria | Exit Criteria | Documentation |
| UAT | * There are no Blocker and Showstopper issues * All the critical functional flows should be working * List of known opens (S1 & S2 issues should not be there) issues is provided to UAT team | * Any Open issues agreed with **Sample Template Project** Technologies should be fixed | * Sign-off Document for acceptance of the development work |

# 

# 6. Testing Phases / Milestones / Resources

|  |  |  |  |
| --- | --- | --- | --- |
| Test Phase | Timeline | Duration | Resources |
| Sprint #0 | Start :  End : |  |  |
| Sprint #1 |  |  |  |
| Sprint #2 | Start : 21/09/2022  End : 7/10/2022 |  |  |
| Sprint #3 |  |  |  |

# 7. Test Environments

The following table lists all environments, owner, and the phase in which each will be used and if there are any compliance requirements.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Environment | Owner | Phase | Compliance |
| QA Environment | QA | Sprints | Unit tests are passing. |
| UAT Environment | Client (**Sample Template Project** Technologies) | UAT | No S1, S2 defects are open |

# 8. Defect Management

The defect management process used to track defects will use Severity. Priority will be used during the stabilization/regression phase or at the end of a project in preparation for the next version.

All defects found during the testing cycle will be logged in JIRA and given a Severity. All fixed defects will be validated in successive drops. If a defect if retested and the functionality is working as expected, the defect will be closed. If a defect is retested but the functionality is not working as expected, the defect will be reopened and assigned back to development team.

|  |  |
| --- | --- |
| Severity | Definition |
| Blocker | Any issue that prevents testing |
| Showstopper | An incident with critical business impact whereby a system is down or immediate work stoppage of a critical business service that threatens current and future productivity, other than for a planned outage to customer or customer’s users |
| Severe | Any reported incident that materially impacts the ability to perform business operations, a workaround is available, however, performance may be degraded, or functions limited. |
| High | An incident where a workaround exists and where there is only limited business impact, or resilience has been lost. |
| Medium | An inquiry for information related to the service that is not directly related to a Problem or incident with no business impact |
| Low | Minor issues that do not impact functionality |

|  |  |
| --- | --- |
| Priority | Definition |
| Severe | To be resolved before all other defects |
| High | To be resolved after all Severe issues have been fixed |
| Medium | To be resolved after all Severe and High issues have been resolved |
| Low | To be resolved after all Severe, High, and Medium issues have been resolved |

The process for reviewing, prioritizing, fixing, promoting fixes back into the testing environment, and re-testing defects is below.

|  |  |
| --- | --- |
| Test Phase | Process |
| Sprints | Development will fix all and/or as many as possible defects filed by QA, in order of Severity (Blockers, Showstoppers, Severe, High, Medium, Low) without need for reviews. |
| Stabilization/Regression | All defects will be triaged at a daily triage meeting and determined what defects must be resolved and what can be assigned to a future release (if applicable) or documented as known issues. This meeting should include representatives from Development, QA, Project Management and Tech Lead. The PM and Tech Lead will assign a Priority to those issues which must be resolved. |
| UAT | All defects will be triaged at a daily triage meeting and determined what defects must be resolved and what can be assigned to a future release (if applicable) or documented as known issues. This meeting should include representatives from Development, QA, Project Management and Tech Lead. The PM and Tech Lead will assign a Priority to those issues which must be resolved. |

# 9. Test Metrics

Below test metrics will be captured as part of testing activities in the project.

## 9.1 User Stories/Defects to be Tested

|  |  |
| --- | --- |
| Name | Purpose |
| Open stories points Vs. closed per week | Trending based on targeted, and actuals of story points completed |
| Open stories points in Stabilization | Trending based on targeted, and actuals of story points completed |
| Open stories in UAT | Trending based on targeted and actuals of # of user story Open. Notes: for example: At time of UAT, if we have any open user story: Red |
| Issues found in Stabilization/Total Issues | -5% of total defects and NO P1/P2 (Green)  +5% of total defects or NO P1 and less than 5 P2 (Yellow)  +5% of total defects with P1 (red) |
| Issues found in UAT/Total Issues | (-5% of total defects and NO P1/P2 (Green)  +5% of total defects or NO P1 and less than 5 P2 (Yellow)  +5% of total defects with P1 (red) |
| Dev. to QA Ratio | Over 2.5 - Red |
|  |  |

## 

## 9.2 Test Cases

|  |  |
| --- | --- |
| Name | Purpose |
| Test Cases Planned per Sprint | Trend the test cases as they have been run to determine how far along QA is in execution of the test cases, based on what was planned. |
| % Test Cases Passed | Trend the % of test cases that have Passed |
| % Test Cases Failed | Trend the % of test cases that have Failed |
| % Test Cases Blocked | Trend the % of test cases that have been blocked |
| % Test Cases Not Yet Run | Trend the % of test cases that haven’t been run |

# 