**TEST PLAN**

**Project “California Marketing”**

**Table of Contents:**

1. Introduction

2. Scope

3. Quality Objectives

4. Test Approach

5. Entry And Exit Criteria

6. Suspension Criteria And Resumption Requirements

7. Test Strategy

8. Resource And Environment Needs

9. Test Plan for Each Testing Type

**INTRODUCTION**

Customers want a perfect website, which passes the full cycle of manual testing. Given the specificity of the site it is important to have the same quality of the site as per requirements.

The Test Plan has been created to facilitate communication within the team

members. This document describe approaches and methodologies that will apply to

the unit, integration and system testing of the https://qasvus.wixsite.com/ca-

marketing. It includes the objectives, test responsibilities, entry and exit criteria,

scope, schedule major milestones, entry and exit criteria and approach. This document

has clearly identified what the test deliverables will be, and what is deemed in and out

of scope.

**2. SCOPE**

The document mainly targets the GUI testing, positive and negative testing, and validating data in report output as per Requirements Specifications provided by Client.

**3. QUALITY OBJECTIVES**

A primary objective of testing is to: assure that the system meets the full requirements, including quality requirements (functional and non-functional requirements) and fit metrics for each quality requirement and satisfies the use case scenarios and maintains the quality of the product. Any changes, additions, or deletions to the requirements document, Functional Specification, or Design Specification will be documented and tested at the highest level of quality allowed within the remaining time of the project and within the ability of the test team.

The secondary objectives of testing will be to identify and expose all issues and associated risks, communicate all known issues to the project team, and ensure that all issues are addressed in an appropriate matter before release. As an objective, this requires careful and methodical testing of the application to first ensure all areas of the system are scrutinized and, consequently, all issues (bugs) found are dealt with appropriately.

**4. TEST APPROACH**

Analytical test approach was used, in accordance with requirements-based strategy, where an analysis of the requirements’ specification form is the basis for planning, estimating and designing tests. Test cases will be created during exploratory testing. All test types are determined in Test Strategy.

The project is using an agile approach, with weekly iterations. At the end of each week the requirements identified for that iteration will be delivered to the team and will be tested.

**5. ENTRY AND EXIT CRITERIA**

**Entry Criteria**

● All test hardware platforms must have been successfully installed, configured, and functioning properly.

● All the necessary documentation, design, and requirements information should be available that will allow

testers to operate the system and judge the correct behavior.

● All test hardware platforms must have been successfully installed, configured, and functioning properly.

● QA resources have completely understood the requirements.

● QA resources have sound knowledge of functionality.

**Exit Criteria**

● No high priority or severe bugs are left outstanding.

● A certain level of requirements coverage has been achieved.

● All high-risk areas have been fully tested, with only minor residual risks left outstanding.

● Reviewed test scenarios, test cases and RTM.

● Cost – when the budget has been spent.

● The schedule has been achieved.

**6. SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS**

**Suspension criteria**

● Software/Hardware problems.

● Significant change in requirements suggested by client.

● The build contains many serious defects which seriously or limit testing progress.

● Assigned resources are not available when needed by the test team.

**Resumption criteria**

● Resumption will only occur when the problem that caused the suspension has been resolved.

**7. TEST STRATEGY**

**QA role in test process:**

● Understanding Requirements.

● Requirement specifications will be sent by client.

**Preparing Test Cases:**

● QA will be preparing test cases based on the exploratory testing. This will cover all scenarios for requirements.

**Preparing Test Matrix:**

● QA will be preparing a test matrix which maps test cases to respective requirements. This will ensure the coverage for requirements.

**Reviewing test cases and matrix:**

● Review for test cases and test matrix will be conducted by QA Lead.

● Any comments or suggestions on test cases and test coverage will be provided by reviewer

● Suggestions or improvements will be updated by preparer and sent to QA Lead for approval.

● Updates and improvements will be reviewed and approved by reviewer.

**Creating Test Data:**

● Test data will be created by respective QA based on scenarios and Test cases.

**Executing Test Cases:**

● Test cases will be executed by respective QA based on designed scenarios, test cases and Test data.

● Test result (Actual Result, Pass/Fail) will be updated in test case document Defect

**Logging and Reporting:**

QA will be logging the defect/bugs in Excel spreadsheet and JIRA, found during execution of test cases.

**Retesting and Regression Testing:**

● Retesting for fixed bugs will be done by respective QA once issue is resolved by respective developer and

bug/defect status will be updated accordingly. In certain cases, regression testing will be done if required.

**Deployment/Delivery:**

● Once all bugs/defect reported after complete testing is fixed and no other bugs are found, report will be deployed to the client, along with sample output by email to respective lead and Report group.

**Bug life cycle:**

● All the issues found while testing will be logged into JIRA.

**Testing types:**

**Manual Testing:**

* Exploratory testing – includes a type of software testing where Test cases are not created in advance but QA checks system “hands-on”. QA may note down ideas about what to test before test execution.
* Positive testing – includes the type of testing that can be performed on the system by providing the valid data as input. It checks whether an application behaves as expected with positive inputs.
* Negative testing – also known as failure testing or error path testing, is a method of testing an application or system that ensures that the plot of the application is according to the requirements and can handle the unwanted input and user behavior. Invalid data is inserted to compare the output against the given input.
* Black box testing – also called Behavioral testing or Partition testing. This kind of testing focuses on the functional requirements of the software. It enables one to derive sets of input conditions that will fully exercise all functional requirements for a program.
* GUI Testing – GUI testing will include testing of the UI part of report. It covers users Report format, look, and feel, error messages, spelling mistakes, GUI guideline violations.
* User acceptance testing - The purpose behind user acceptance testing is to confirm that system is developed according to the specified user requirements and is ready for operational use. Acceptance testing is carried out at two levels - Alpha and Beta Testing. User acceptance testing (UAT) will be done at the Client.
* Boundary Testing - The process of testing between extreme ends or boundaries between partitions of the input values.
* ADHOC testing – includes an informal testing type with an aim to break the system.

**Automation Testing:**

Functional Automation testing - testing of website’s interface and functionality using Selenium Webdriver. Functional testing is carried out in order to find out unexpected behavior of the report. The purpose of functional testing is to provide correctness, reliability, testability and accuracy of the report output/data.

API testing - testing of correct form of API requests, JSON body using Postman API.

Performance Automation testing - testing of website’s performance with Lighthouse, GTMetrix.

Performance Testing:

* Check the optimal time the page is loaded
* Check the operation of the system under load

Automation Security testing - a type of software testing that intends to uncover vulnerabilities of the system and determine that its data and resources are protected from possible intruders. Testing of website’s performance with Observatory Mozilla.

**8. RESOURCES AND ENVIRONMENT NEEDS**

**Testing Tools:**

● Process Tool

● Test case creation Microsoft Word, Microsoft Excel, JIRA

● Test case tracking JIRA, Confluence

● Test case execution Manual

● Test case management Microsoft Excel, JIRA, Confluence

● Test reporting JIRA

● Check list creating Microsoft Excel, JIRA

**Test Environment x Support level 1 (browsers):**

● Windows 11

● Chrome 97

● Firefox 95

1. **TEST PLAN FOR EACH TESTING TYPE IN THIS PROJECT**
2. Manual Testing:

Positive Testing, Cross-Browser Testing: Within 5 minutes we are going to verify the main functionality of the given website, such as:

* Verify website link.
* Validate that "Twitter" icon is functional and leads to the twitter.com website.
* Verify the completion of the “Subscribe Form Email address” is correct.
* Validate that "shopping cart" is functional.
* Verify visibility of text in section "About" on the Homepage.
* Verify that the video images in the right format in "Video" section on the Homepage.
* Verify that the video images are aligned.
* Verify that the video section contains eight videos (four videos in the first row and four in the second row).
* Verify that the '"contact form" is readable.
* Validate "Submit" button in contact form on Homepage is functional and clickable after required field is filled.
* Verify that "Submit" button in contact form on Homepage doesn't contain any misspelling.

Negative Testing: (approximate time - 3 minutes)

* Verify that invalid email Ghihg@ghigk.com entered in subscribe form is not accepted.
* Verify that the user can not enter 0 in the quantity field.
* Verify that invalid email 123456@gmail.com entered in subscribe form is not accepted.

Boundary Testing: (approximate time - 3 minutes)

* Verify that you can buy products quantity of "0" pieces
* Verify that you can buy products quantity of "5000" pieces.
* Verify that you can buy products quantity of "99999" pieces
* Verify that you can buy products quantity of "100000" pieces

Ad-Hoc: (approximate time - 3 minutes)

* Verify that the button "learn more" on Homepage leads to the wrong website
* Verify that "email" field in section "Connect with us" on Homepage using special character "#" 5 times is invalid
* Verify that using special character "%" 3 times in "Search" in section "Blog" (main menu) is invalid

Equivalence Partitioning Testing: (approximate time – 5 minutes)

* Verify that you can enter test data in the "City" field – AT7
* Verify that entered test data in the "City" field is invalid – AT7@
* Verify that entered test data in the "City" field is invalid – AT7@<
* Verify that entered test data in the "City" field is invalid – AT7@<?space
* Verify that entered test data in the "City" field is invalid – AT7@<? AT
* Verify that entered test data in the "City" field is invalid - AT7@<? AT7
* Verify that entered test data in the "City" field is invalid - AT7@<? AT7@

1. Automation Testing. Within 2 hours we are going to write the script for the given website, to verify the main functionality, such as:

* Google opens the website for **https://qasvus.wordpress.com/.**
* Verify website link
* Validate and print the page title.
* Locate the name, email and message fields
* Fil out all fields: name, email, message
* Submit the message
* Go back to the Homepage page

1. API Testing. Within 2 hours we are going to write the script for the given website, to verify the main functionality, such as:

* Create a new address at the <https://qasvus.wixsite.com/ca-marketing/account/my-addresses>.
* Get information about all the addresses on the page.
* Get the specific address which was created.
* Modify the specific address.
* Get the modified address and check it.
* Delete the modified address.
* Get the updated address.

1. Performance Automation Testing with Lighthouse and GTMetrix:

* We are getting a general performance outlook of the [https://qasvus.wixsite.com/ca-marketing](https://qasvus.wixsite.com/ca-marketing/account/my-addresses) website using Lighthouse Chrome extension and GTMetrix tool. Approximate time - 5 minutes.

1. Automation Security Testing: (approximate time - 3 minutes)

* We are testing security of the [https://qasvus.wixsite.com/ca-marketing](https://qasvus.wixsite.com/ca-marketing/account/my-addresses) using the website scanner tool at the <https://observatory.mozilla.org/> website.