Test Plan

Project “California Marketing”

Table of Contents:

1. INTRODUCTION

The Test Plan outlines the scope, approach, resources, and schedule for all testing activities related to the California Marketing website project. The website can be accessed at https://qasvus.wixsite.com/ca-marketing .

The plan defines what will be tested, including the specific items and features that will undergo testing. It also specifies the types of testing that will be conducted. The responsible personnel for carrying out the testing are identified, along with the necessary resources and timeframe required to complete the testing process. Additionally, the plan addresses the potential risks associated with the testing activities.

The main objective is to deliver a flawless website that has undergone a comprehensive cycle of manual testing. Considering the unique nature of the website, maintaining the same level of quality is of paramount importance to meet the customer's expectations.

2. SCOPE

The document targets GUI testing, API, security and performance testing of client's data in report output as it was previously discussed at the meeting.

- ➢ GUI
- ➢ Performance
- ➢ Security
- ➢ API Server response

MANUAL EXPLORATORY TESTING

PURPOSE: The purpose of this test is to make sure critical defects are removed before the next levels of testing can start.
SCOPE: Includes various aspects of the website, such as the website link, sign-up functionality, log-in functionality, integration with Vk (a social media platform), shopping cart, connect-with-us form, and registration form. These areas will be the primary focus of the exploratory testing.
TESTERS: Testing team.
METHOD: This exploratory testing is carried out in the application without any test scripts and documentation.
TIMING: At the beginning of each cycle.

WEBSITE MANUAL TEST

PURPOSE: Functional testing will be performed to check the functions of the application. The functional testing is carried out by feeding the input and validates the output from the application.
SCOPE: The below table details the scope of Functional test.
TESTERS: Testing Team.
METHOD: The test will be performed according to Functional scripts.
ENVIRONMENT: Windows 10 : Edge(Version 114.0.1823.58), Chrome (Version 114.0.5735.199).
TIMING: After the Exploratory test is completed.

The following test cases covers the manual tests:

Test Case ID	***Business Requirement***	***Expected Results***	***Browsers***	***Actual Results***	***Status***
POSITIVE TESTS					
TC - 001	Verify Website Link	Link should open on Home page	Google Chrome	Link opened on Home page	PASS
			Microsoft Edge	Link opened on Home page	PASS
TC - 002	Sign-up on the website https://qasvus.wixsite.com/ca-marketing	Create a new account	Google Chrome	Created a new account	PASS
			Microsoft Edge	Created a new account	PASS
TC - 003	Verify login with valid username and password	User got in the account	Google Chrome	User got in the account	PASS
			Microsoft Edge	User got in the account	PASS

TC - 004	Validate Vk linked icon leads to correct page	Website link should open on Vk	Google Chrome	Website link opened on Vk	PASS
			Microsoft Edge	Website link opened on Vk	PASS
TC - 005	Add to shopping cart a product and checkout	Add product in shopping cart and checkout	Google Chrome	Added product in shopping cart but couldn't checkout	FAIL
			Microsoft Edge	Added product in shopping cart but couldn't checkout	FAIL
TC - 006	Verify “Connect With Us” form	Submitting form was successful	Google Chrome	Submitting form was unsuccessful	FAIL
			Microsoft Edge	Submitting form was unsuccessful	FAIL
NEGATIVE TESTS					
TC - 001	Verify Registration for event with invalid first and last name	Not allowed to register with invalid first and last name	Google Chrome	Registration was allowed with invalid first and last name	FAIL
			Microsoft Edge	Registration was allowed with invalid first and last name	FAIL
TC - 002	Adding to shopping cart 0 value	Enter minimum amount of 1	Google Chrome	Enter minimum amount of 1	PASS
			Microsoft Edge	Enter minimum amount of 1	PASS
TC - 003	Sign-up on website with invalid email and valid password	Sign-up with invalid email is not allowed	Google Chrome	Sign-up with invalid email is not allowed	PASS
			Microsoft Edge	Sign-up with invalid email is not allowed	PASS

WEBSITE AUTOMATION TEST

PURPOSE: This test focuses on creating automation scripts based on manual test cases.
TESTERS: Testing Team.
SCOPE: The same as in Manual Functional Testing.
TOOLS: The testing team should use PyCharm as a main Integrated Development Environment (IDE), Python as the primary programming language, Selenium WebDriver as the primary framework for automating test scripts and BrowserStack to run tests in various environments.
ENVIRONMENT: Windows 10 : Edge(Version 114.0.1823.58).
TIMING: After manual testing is done and all critical issues are resolved.

WEBSITE API TEST

PURPOSE: API tests are performed to determine if the developed APIs meet the expectations when it comes to the functionality, performance, reliability and security of the website.
TESTERS: Testing Team.
SCOPE: https://qasvus.wixsite.com/ca-marketing.
TOOLS: The testing team should use Postman API as a platform for building and using APIs and JavaScript as a language for test scripts.
TIMING: After manual testing is done and all critical issues are resolved.

The following test cases covers API tests:

Test Case ID	***Requests***	***Expected Results***	***Actual Results***	***Status***
POSITIVE TESTS				
TC - 001	POST /Add new postal address	New address is added Status code is 200	New address is added Status code is 200	PASS

TC - 002	POST / Make default address	New address is set as default Status code is 200	New address is set as default Status code is 200	PASS
TC - 003	GET / Show default address	Default address is showing Status code is 200	Default address is showingStatus code is 200	PASS
TC - 004	GET / Show all addresses	List of all addresses is showing Status code is 200	List of all addresses is showing Status code is 200	PASS
TC - 005	GET / Show specific address	Specific address is showing Status code is 200	Specific address is showing Status code is 200	PASS
TC - 006	PATCH / Modify existing address	New address is modified Status code is 200	New address is modified Status code is 200	PASS
TC - 007	GET / Check updated postal address	Modified address is showing Status code is 200	Modified address is showing Status code is 200	PASS
TC - 008	DEL / Delete postal address	Address is deleted Status code is 200	Address is deleted Status code is 200	PASS
NEGATIVE TESTS				
TC - 001	DELETE / Delete address second time	Can’t find address by id Status code is 400	Can’t find address by id Status code is 400	PASS

TC - 002	GET / Check deleted postal address	Can't find address by id Status code is 400	Can't find address by id Status code is 400	PASS
TC - 003	POST / Make default deleted address	Can't find address by id Status code is 400	Can't find address by id Status code is 400	PASS
TC - 004	Patch / Update deleted address	Can't find address by id Status code is 400	Can't find address by id Status code is 400	PASS

WEBSITE AUTOMATION PERFORMANCE TEST

PURPOSE: This test is performed to measure the speed, responsiveness, stability of the site and also how well the page is built for optimal performance.
TESTERS: Testing Team.
SCOPE: https://qasvus.wixsite.com/ca-marketing.
TOOLS: The testing team will use Google Lighthouse, GTMetrix, Webpagetest. These tools provide insights into various performance metrics and help analyze the performance of web pages.
METRICS: Page Load, Speed Index, FCP (first content paint), LCP (largest content paint), TBT (total blocking time), CLS (cumulative layout shift).
TIMING: After manual testing is done and all critical issues are resolved.

WEBSITE AUTOMATION SECURITY TEST

PURPOSE: This test is performed to reveal current or potential security vulnerabilities.
TESTERS: Testing Team.
SCOPE: https://qasvus.wixsite.com/ca-marketing.
TOOLS: The testing team will use Mozilla Observatory and ImmuniWeb.
TIMING: After manual testing is done and all critical issues are resolved.

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 maintain the quality of the product.

At the end of the project development cycle, the user should find that the project has met or exceeded all of their expectations as detailed in the requirements.

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 manner before release.

As an objective, this requires careful and methodical testing of the website to first ensure all areas of the system are scrutinized and, consequently, all issues (bugs) found are dealt with appropriately.

4. TEST APPROACH

We will be using an analytical approach, which means we will carefully analyze the requirements specification to plan, estimate, and design our tests. Test cases will be created during exploratory testing, although many of them may already be predetermined in our Test Strategy.

The project follows an agile approach with weekly iterations. At the end of each week, we will receive the requirements for that specific iteration. We will then test those requirements to ensure they are functioning as expected. This iterative process allows for continuous feedback and adjustment throughout the development and testing stages.

5. ENTRY AND EXIT CRITERIA

The ***Entry criteria*** specify the requirements that must be fulfilled before starting testing:

- Proper test data should be available.

- 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 the standard software tools including the testing tools must have been successfully installed and functioning properly.
- All test hardware platforms must have been successfully installed, configured, and functioning properly.

The ***Exit criteria*** are the conditions that must be satisfied in order to proceed with the implementation phase:

- 100% Test Scripts executed
- 95% pass rate of Test Scripts
- No open Critical and High severity defects
- 95% of Medium severity defects have been closed
- All remaining defects are either canceled or documented as Change Requests for a future release
- All expected and actual results are captured and documented with the test script
- All defects logged in Jira
- Cost – when the budget has been spent.
- The schedule has been achieved.

6. SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

Suspension criteria are predefined conditions or situations that trigger a temporary halt or suspension of the testing process. These criteria indicate when it is necessary to pause testing activities due to certain circumstances or issues.

- 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 are the conditions that need to be met in order to resume testing after a suspension. They ensure that any underlying issues or obstacles have been adequately resolved, allowing for a smooth continuation of the testing process.

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 the preparer and sent to QA Lead for approval.
- Updates and improvements will be reviewed and approved by the 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:

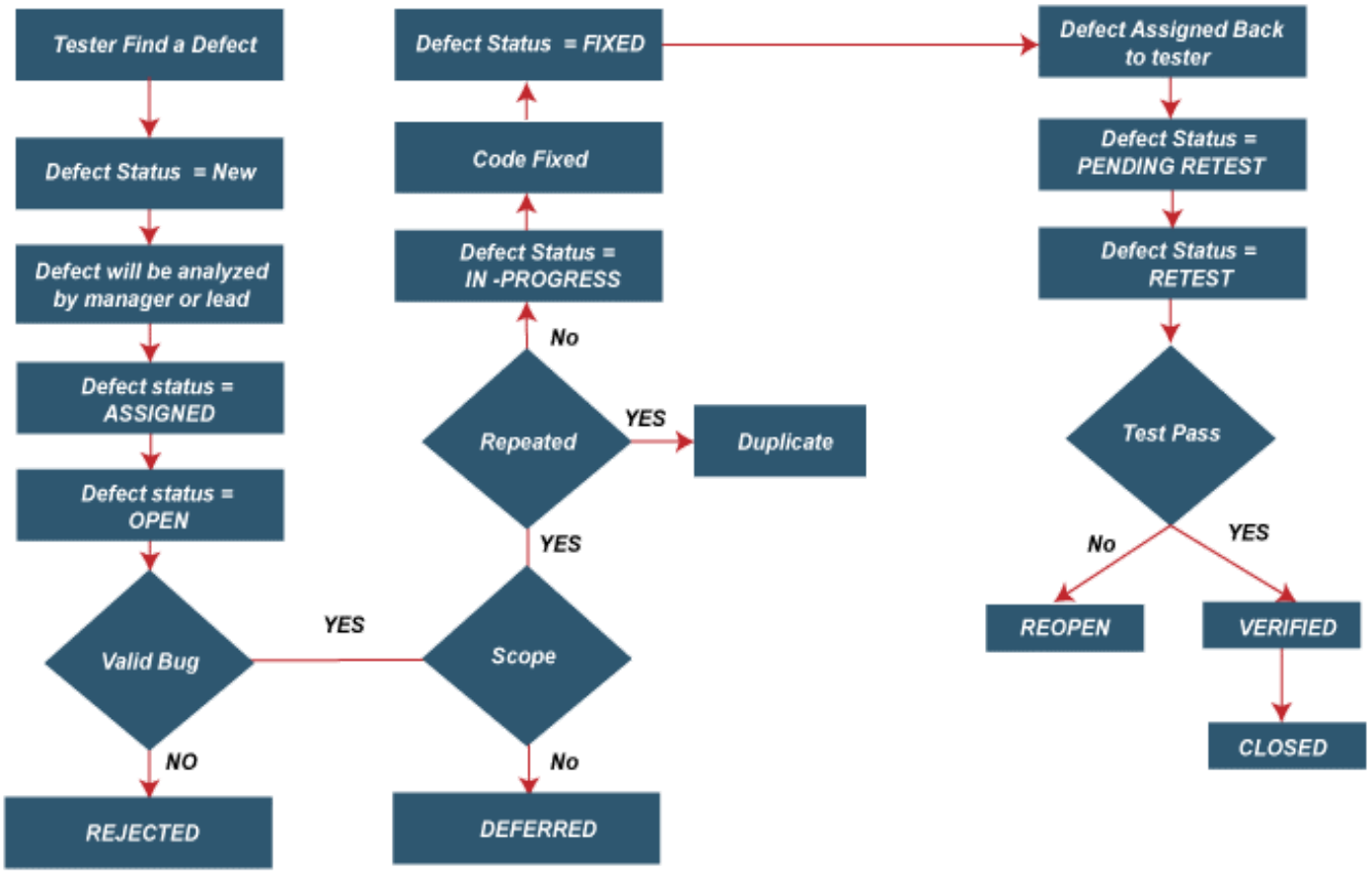
- Retesting for fixed bugs will be done by respective QA once issue is resolved by respective developer and bug/defect status will be updated accordingly. In certain cases, regression testing will be done if required.

Deployment/Delivery:

- Once all bugs/defects reported after complete testing are fixed and no other bugs are found, a 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.



Bug Severity and Priority Definition

Bug Severity and Priority fields play crucial roles in categorizing and prioritizing bugs for resolution. The severity level of a bug indicates the impact and criticality of the issue, while the priority level determines the order in which bugs will be addressed and fixed.

During testing, the testing team will assign a severity level to each bug, ensuring that the appropriate severity is accurately assigned. The Test Lead holds the responsibility for overseeing this process and ensuring correct severity levels are assigned.

Bug Triage Meetings will be conducted with the participation of the QA Lead, Development Lead, and Project Manager. These meetings serve the purpose of reviewing all active bugs and collectively determining their priority. The QA Lead will organize these routine meetings to address both new and existing unresolved bugs.

Severity List

Severity ID	Severity	Severity Description
1	Highest	The module/product crashes or the bug causes non-recoverable conditions. System crashes, or database or file corruption, or potential data loss, program hangs requiring reboot are all examples of a Severity 1 bug.
2	High	Major system components unusable due to failure or incorrect functionality. Severity 2 bugs cause serious problems such as a lack of functionality, or insufficient or unclear error messages that can have a major impact on the user, prevent other areas of the app from being tested, etc. Severity 2 bugs can have a work around, but the work around is inconvenient or difficult.
3	Medium	Incorrect functionality of component or process. There is a simple work around for the bug if it is Severity 3.
4	Low	Documentation errors or signed off Severity 3 bugs.

Priority List

Priority	Priority Level	Priority Description
1	Highest	This bug must be fixed immediately; the product cannot ship with this bug.
2	High	These are important problems that should be fixed as soon as possible. It would be an embarrassment to the company if this bug shipped.
3	Medium	The problem should be fixed within the time available. If the bug does not delay the shipping date, then fix it.
4	Low	It is not important (at this time) that these bugs be addressed. Fix these bugs after all other bugs have been fixed. Enhancements/ Good to have features incorporated- just are out of the current scope.
5	Lowest	Documentation errors or signed off Low 4 bugs.

Testing types:

Exploratory testing :

Exploratory testing will include a type of software testing where Test cases are not created in advance but QA check system on the fly. QA may note down ideas about what to test before test execution.

GUI Testing :

GUI testing will include testing of the UI part of the report. It covers users Report format, look and feel, error messages, spelling mistakes, GUI guideline violations.

Positive testing:

Positive testing will include 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:

Negative testing will include a method of testing an application or system that ensures that 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. Negative testing is also known as failure testing or error path testing. When performing negative testing errors messages are expected.

ADHOC testing:

ADHOC testing will include an informal testing type with an aim to "break" the system.

Smoke Testing:

Smoke Testing is a software testing process that determines whether the deployed software build is stable or not. Smoke testing is a confirmation for the QA team to proceed with further software testing. It consists of a minimal set of tests run on each build to test software functionalities. Smoke testing is also known as "Build Verification Testing" or "Confidence Testing." In simple terms, we are verifying whether the important features are working and there are no showstoppers in the build that is under testing.

Functional Testing:

Functional testing is carried out in order to find out unexpected behavior of the report. The characteristics of functional testing are to provide correctness, reliability, testability and accuracy of the report output/data.

Boundary testing:

Testing technique that focuses on testing the boundaries or limits of input values. It aims to uncover any issues or vulnerabilities that may occur at the edges of the input domain, such as minimum and maximum values, as well as values near the boundaries.

API testing:

API testing is a type of software testing that analyzes an application program interface (API) to verify it fulfills its expected functionality, security, performance and reliability.An API test is generally performed by making requests to one or more API endpoints and comparing the response with expected results.

Performance testing:

Performance testing is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload. Performance tests are typically executed to examine speed, robustness, reliability, and application size.

8. RESOURCES AND ENVIRONMENT NEEDS

Testing Tools:

Process	Tool
Test case creation	Microsoft Word, Google Sheets, JIRA
Test case tracking	JIRA, Confluence
Test case execution	Manual, Selenium WebDriver, Mozilla Observatory, Lighthouse, GTMetrix, Postman, BrowserStack
Test case management	Google Sheets, JIRA, Confluence
Test reporting	JIRA
Check list creating	Google Sheets, JIRA
API Testing	Postman
Performance testing	Lighthouse, GT Metrix, WebPage test
Automation Testing	Selenium WebDriver, XPath (for locators searching)

Test Environment (browsers):

Windows 10 : Edge(Version 114.0.1823.58), Chrome (Version 114.0.5735.199)

TEST SCHEDULE

Task Name	Start	Finish	Effort	Comments

APPROVALS

	Project Manager	Scrum Master	Evaluation Manager
Name			
Signature			

TERMS/ACRONYMS

The below terms are used as examples, please add/remove any terms relevant to the document.

TERM/ACRONYM	DEFINITION
API	Application Program Interface
GUI	Graphical user interface
PM	Project manager
UAT	User acceptance testing
CM	Configuration Management
QA	Quality Assurance
RTM	Requirements Traceability Matrix