TEST PLAN

California Marketing
https://qasvus.wixsite.com/ca-marketing

Table of Contents

1. INTRODUCTION

The test plan was created to facilitate communication between team members. This document provides guidance on planned works to be conducted and terms. This document describes the approaches and methodologies that will be applied to the modules: "Track Artist" button, "Facebook Icon" , "I'm a product 12" quantity button "Log In" verification , and Phone number in the "My addresses" section of the website https://qasvus.wixsite.com/ca-marketing. The different types and approaches will be applied during testing. The testing will be based on technical requirements.

2. SCOPE

The document mainly targets the Positive, Negative, Boundary, Equivalence Partitioning, Automation Performance and Automation Security testing of the "Track Artist" button, "Facebook Icon" , "I'm a product 12" quantity button "Log In" verification , and Phone number in the "My addresses" section of the website https://qasvus.wixsite.com/ca-marketing.

3. ENTRY CRITERIA

- Test documentation is created.

- The test environment is ready.

- Server for testing is available.

4. DELIVERABLES

- Checklists.

- Bug reports in Jira.

- Improvements in Jira.

5. EXIT CRITERIA

- All checks are passed.

- All bugs in Jira are fixed or improved (blocker or critical severity defects are fixed).

- Planned deliverables are ready.

- Project is approved and ready for production by: QA engineer, PM/BA.

6. TEST EXECUTION

6.1 WEBSITE MANUAL TEST

Execute manual testing for the "Track Artist" button, "Facebook Icon" , "I'm a product 12" quantity button "Log In" verification , and Phone number in the "My addresses" section of the website https://qasvus.wixsite.com/ca-marketing based on the next types of testing:

- Exploratory
- Positive testing.
- Negative testing.
- Boundary testing.
- Equivalence Partitioning.

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.

Functional Positive testing:
- Checks whether an application behaves as expected with positive inputs.

Functional Negative testing:
- Invalid data is inserted to compare the output against the given input.

Boundary testing:
- Boundary testing aims to check whether there are any errors at the boundaries of an input domain and helps developers ensure that their software contains minimal errors before officially releasing it to the public.

Equivalence Partitioning:
- Equivalence partitioning divides input domain into classes of data, and with the help of these classes of data, test cases can be derived.

Environment Support:

- Laptop
- OS: Windows 11 Home, 64-bit operating system, x64-based processor
- Browsers (latest versions): Microsoft Edge

The following test cases are covering the Website's manual tests:

Positive Testing

TC ID:	Test Action	Expected results	Pass/Fail
TC - 001	Verify that "Track Artist" button is clickable	A window with instructions how to track an artist appears	Pass
TC - 002	Verify that "Track Artist" log in option works	The user is logged in using the instructions on the Website	Pass
TC - 003	Verify that "Facebook Icon" redirects a user to the Facebook Page of a Media or Music company	The user goes to the Media or Music company Facebook page	Fail

Negative Testing

TC ID:	Test Action	Expected results	Pass/Fail

TC – 004	Verify that section "Artist Name" is static if using multiple clicks on it	section "Artist Name" is static if using multiple clicks on it	Pass
TC - 005	Verify that user cannot enter 0 in the quantity button "I'm a product 12"	Enter a minimum amount of 1	Pass
TC - 006	Double check your email or password and try again	Double check your email or password and try again	Fail

Boundary Testing

TC ID:	Test Action	Expected results	Pass/Fail
TC – 007	Verify that user cannot enter 0 in the quantity button "I'm a product 12"	User enters a minimum amount of 1	Pass
TC - 008	Verify that user can enter 1 in the quantity button "I'm a product 12"	user can pick 1 white item product 12	Pass
TC - 009	Verify that user can enter 99998 in the quantity button "I'm a product 12"	user can pick 99998 white items product 12	Pass

Equivalence Partitioning Testing

TC ID:	Test Action	Expected results	Pass/Fail
TC – 007	Verify the phone number in the "My addresses" included 10 integers is valid	user logs in to account	Pass
TC - 008	Verify the phone number in the "My addresses" included 11 integers is invalid	Please enter valid phone number notification should appear	Fail
TC - 009	Verify the phone number in the "My addresses" included 9 integers is invalid	Please enter valid phone number notification should appear	Fail

Testing Tools used:

Process	Tools

Test case creation	Microsoft Word, Microsoft Excel, JIRA
Test case tracking	Microsoft Word, Microsoft Excel, JIRA
Test case execution	Manual
Test case management	Microsoft Excel, JIRA
Test reporting	Microsoft Excel, JIRA

6.2. WEBSITE AUTOMATION TEST

Automatization of our manual testing by using the **Selenium IDE**. It will help automate the testing process and reduce the need for manual testing by examining the "Track Artist" button, "Facebook Icon" , "I'm a product 12" quantity button "Log In" verification , and Phone number in the "My addresses" section of the website https://qasvus.wixsite.com/ca-marketing.

Environment Support:

- Laptop
- OS: Windows 11 Home, 64-bit operating system, x64-based processor
- Browsers (latest versions): Microsoft Edge

TC ID:	Test Action	Expected results	Pass/Fail
TC - 001	Verify that "Track Artist" button is clickable	A window with instructions how to track an artist appears	Pass
TC - 002	Verify that "Track Artist" log in option works	The user is logged in using the instructions on the Website	Pass
TC - 003	Verify that "Facebook Icon" redirects a user to the Facebook Page of a Media or Music company	The user goes to the Media or Music company Facebook page	Fail

6.3 WEBSITE API TEST

The plan for the API testing is to create requests to POST, GET and DELETE address on our account on https://qasvus.wixsite.com/ca-marketing website. We need to fully inspect response from the server by creating prescripts, tests, and the appropriate environment. 1) Create the POST request which creates a new address and additional tests that check the response. Create the GET request which fetches the info about the newly created address. Create a PATCH request that makes changes to the address. Create the DELETE request that deletes this new address. Create the DELETE request again to check if the address is deleted indeed. Then create the GET request to make sure that we cannot fetch the deleted address.

The following test cases are covering API tests:

Positive Testing

TC	Request / Test Action	Expected Results	Actual Results	Pass/ Fail
TC-001	POST / New address with correct values is entered.	New address is added. Status code is 200	New address is added. Status code is 200	Pass
TC-002	POST / Set up a new address as default	The new address is set as default. Status code is 200	The new address is set as default. Status code is 200	Pass
TC-003	GET / Show all addresses	List of all addresses is shown. Status code is 200	List of all addresses is shown. Status code is 200	Pass
4	GET / Show default address	Default address is shown. Status code is 200	Default address is shown. Status code is 200	Pass
5	PATCH / Modify new address info cells.	New address is modified. Status code is 200	New address is modified. Status code is 200	Pass

6	GET / Show modify address	Modify address is shown. Status code is 200	Modify address is shown. Status code is 200	Pass
7	DEL / Delete new address	New address is deleted. Status code is 200	New address is deleted. Status code is 200	Pass
8	DEL / Delete new address 2nd time	The new address has already been removed. Status code is 400	The new address has already been removed. Status code is 400	Pass
9	GET / Check deleted new address	The deleted address is not in the list. Status code is 400	The deleted address is not in the list. Status code is 400	Pass
Negative Testing				
10	POST / Address with incorrect First Name is entered (numbers)	Address with Incorrect First Name is not added. Status code is 400	Address with Incorrect First Name is added. Status code is 200	Fail
11	GET / Check entered address with incorrect First Name (numbers)	The address is not found. Status code is 404	The address is in the list. Status code is 200	Fail
12	DEL / Delete entered address with incorrect First Name (numbers)	The address is not found Status code is 404	The address is deleted. Status code is 200	Fail
13	POST / Address with incorrect First Name is entered (special characters)	Address with Incorrect First Name is not added. Status code is 400	Address with Incorrect First Name is added. Status code is 200	Fail
14	GET / Check entered address with incorrect First Name (special characters)	The address is not found. Status code is 404	The address is in the list. Status code is 200	Fail

15	DEL / Delete entered address with incorrect First Name (special characters)	The address is not found. Status code is 404	The address is deleted. Status code is 200	Fail
16	POST / Address with incorrect First Name is entered (empty)	Address with Incorrect First Name is not added. Status code is 400	Address with Incorrect First Name is added. Status code is 200	Fail
17	GET / Check entered address with incorrect First Name (empty)	The address is not found. Status code is 404	The address is in the list. Status code is 200	Fail
18	DEL / Delete entered address with incorrect First Name (empty)	The address is not found. Status code is 404	The address is deleted. Status code is 200	Fail
19	POST / Address with informal First Name is entered (a lot of the same letters)	Address with informal First Name is not added. Status code is 400	Address with informal First Name is added. Status code is 200	Fail
20	GET / Check entered address with informal First Name (a lot of the same letters)	The address is not found. Status code is 404	The address is in the list. Status code is 200	Fail
21	DEL / Delete entered address with informal First Name (a lot of the same letters)	The address is not found. Status code is 404	The address is deleted. Status code is 200	Fail

6.4 WEBSITE AUTOMATION PPERFORMANCE TEST

Performance automation tests execute for measures the speed, responsiveness, and stability of the tested website. Tests are executed in incognito environments.

After testing reports were generated.

Testing Tools used:

Process	Tools
Test case creation	Google Lighthouse, GTMetrix, BrowserStack-SpeedLab
Test case tracking	Google Lighthouse, GTMetrix, BrowserStack-SpeedLab
Test case execution	Automation
Test reporting	txt -, html -, pdf -, png - files

6.5 WEBSITE AUTOMATION SECURITY TEST

Security automation tests are executed to reveal potential flaws or weaknesses of software and the website. Testing focuses on whether the application is designed and configured correctly.

Tests are executed in incognito environments.
After testing reports were generated.

Testing Tools used:

Process	Tools
Test case creation	Mozilla Observatory, ImmuniWeb
Test case tracking	Mozilla Observatory, ImmuniWeb
Test case execution	Automation
Test case management	
Test reporting	txt -, html -, pdf - files

7. DEFECT MANAGEMENT

Defects found during testing will be logged in Jira, including details such as steps to reproduce, severity, and priority. The development team will be responsible for resolving the defects, and the QA team will retest the fixes before closing the defects.

8. CONCLUSION

This Test Plan outlines the testing approach and activities for the website. By following this plan, the QA team will ensure that the website meets the specified

requirements, functions properly across different browsers and devices, and provides a positive user experience for users.