**TEST PLAN**

**Project “CALIFORNIA MARKETING”**

<https://qasvus.wixsite.com/ca-marketing>

Test Plan created by: Dmitry Pavlov Project Manager: Sergey Efremov

**Table of Contents:**

1. INTRODUCTION

2. SCOPE

3. 3. SUB-TASKS

3.1 Manual test

3.2 Automation test

3.3 API test

3.4 Performance Automation test

3.5 Automation Security test

4. TESTS SCHEDULE AND ESTIMATED TIME

**1. INTRODUCTION**

The Test Plan has been created to facilitate communication within the team members. This document describes approaches and methodologies that will apply to the modules: “User Account registration and login”, “Chat”, “Artist name (upcoming shows)”, “Subscribe Form”, “Products” and “Connect with us” of the website <https://qasvus.wixsite.com/ca-marketing>.

This document has clearly identified sub-tasks of website's module testing, types of testing, what devices, environments and tools to be used. It includes test cases with expected and actual results, schedule and executing estimated time each of sub-tasks and entire project.

**2. SCOPE**

The document mainly targets the GUI, Exploratory, Functional Positive, Negative, Ad-hoc, Boundary and Equivalence Partitioning testing of User’s Personal Account Menu, Performance and Security testing of the website.

**3. SUB-TASKS**

3.1 Manual test

Execute manual testing for the User’s Personal Account Menu based on the next types of testing:

* GUI
* Exploratory testing
* Positive testing
* Negative testing
* Ad-hoc testing
* Boundary testing
* Equivalence Partitioning testing

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.

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.

Ad-hoc testing – includes an informal testing type with an aim to break the system.

Boundary testing – is the process of testing between extreme ends or boundaries between partitions of the input values.

Equivalence Partitioning testing – is type of black box testing technique where input data units are divided into equivalent partitions that can be used to derive test cases which reduces time required for testing because of small number of test cases.

Test Environment and Support (browsers):

● Laptop: Lenovo Inspiration 3793

● OS: Windows 11/64

● Google Chrome, version 114.0.5735.91 (Official Build) (64-bit)

● Brave, version 1.51.118 (64-bit)

● Microsoft Edge, version 113.0.1774.57 (64-bit)

● Mozilla Firefox, version 113.0.2 (64-bit)

● Opera, version 99.0.4788.31

The following test cases are covering next manual tests:

|  |  |  |
| --- | --- | --- |
| Test Case ID | Test requirements | Expected Test Result |
| Positive testing | | | |
| 001 | Verify Sign up for new user account with email | User signed up; a new account created |
| 002 | Verify login with valid username and password | The user logged in the account |
| 003 | Verify Chat functionality | The user is able to chat, leave messages, insert icons (smiles etc.), attach files |
| 004 | Verify user is able to get notifications for upcoming show | The user get notification |
| 005 | Verify user is able to proceed to Facebook and Twitter pages in "Artist name" block | The user proceeds to Facebook and Twitter pages |
| 006 | Verify Map functionality | The user can use all the functionality of the Map section |
| Negative testing | | | |
| 007 | Verify that invalid email id and valid password is entered | Double check your email and try again |
| 008 | Verify that invalid email id and invalid password is entered | Double check your email or password and try again |
| 009 | Verify the completion of the Subscribe Form Email address using same positive integer is invalid | Please fill in a valid email address |
| 010 | Verify that user can not enter 0 in the quantity button "I'm a product 12" | Enter a minimum amount of 1 |
| 011 | Verify section Artist Name is static if using multiple clicks on it | Section “Artist Name” is static if using multiple clicks on it |
| Ad-hoc testing | | | |
| 012 | Verify the Subscribe Form using same positive integer is invalid | Please fill in a valid email address |
| 013 | Verify the Quantity box of "I'm a product3" doesn't exceed left in stock by entering massive number | Only 99999 lefts in stock |
| 014 | Verify Email field does not allow to enter incomplete email | Email field became red with alert "Please enter '@'. 'qwerty@' is incomplete |
| Boundary testing | | | |
| 015 | Verify that user can not enter 0 in the quantity button "I'm a product 12" | Enter a minimum amount of 1 |
| 016 | Verify that user can enter 1 in the quantity button "I'm a product 12" | User can pick 1 white item product 12 |
| 017 | Verify that user can enter 2 in the quantity button "I'm a product 12" | User can pick 2 white items product 12 |
| 018 | Verify that user can enter 99998 in the quantity button "I'm a product 12" | User can pick 99998 white items product 12 |
| 019 | Verify that user can enter 99999 in the quantity button "I'm a product 12" | User can pick 99999 white items product 12 |
| 020 | Verify that user can not enter 100000 in the quantity button "I'm a product 12" | Only 99999 lefts in stock |
| Equivalence Partitioning testing | | | |
| 021 | Verify the phone number in the “My addresses" included 10 integers is valid | User go to account |
| 022 | Verify the phone number in the “My addresses" included 11 integers is invalid | Please enter valid phone number |
| 023 | Verify the phone number in the “My addresses" included 9 integers is invalid | Please enter valid phone number |

Testing Tools used:

|  |  |
| --- | --- |
| **Process** | **Tools** |
| Test case creation | MS Word, MS Excel, JIRA |
| Test case tracking | MS Word, MS Excel, JIRA |
| Test case execution | Manual |
| Test case management | MS Excel, JIRA |
| Test reporting | MS Excel, JIRA |

3.2 Automation test

Automation tests are created by using SeleniumIDE and Selenium Python UnitTest frameworks based on Functional manual test cases and executes to check functionality of User’s Personal Account Menu.

The following test cases are covering automation tests (SeleniumIDE):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Test requirements | Expected Test Result | Actual Test Results | Pass/Fail |
| Positive testing | | | | | |
| 002 | Verify login with valid username and password | User logged in the account | User logged in the account | Pass |
| 003 | Verify Chat functionality | User is able to chat, leave messages, insert icons (smiles etc.), attach files | User is able to chat, leave messages, insert icons (smiles etc.), attach files | Pass |
| 004 | Verify user is able to get notifications for upcoming show | User get notification | User did not get notification. | Fail |
| 005 | Verify user is able to proceed to Facebook and Twitter pages in "Artist name" block | User proceeds to Facebook and Twitter pages | User did not proceed to Facebook and Twitter pages. | Fail |
| Negative testing | | | | | |
| 007 | Verify that invalid email id and valid password is entered | Double check your email and try again. | Double check your email and try again. | Pass |
| 008 | Verify that invalid email id and invalid password is entered | Double check your email or password and try again. Passwords must be at least 4 characters long. Try again. | Double check your email or password and try again. Passwords must be at least 4 characters long. Try again. | Pass |
| 009 | Verify the completion of the Subscribe Form Email address using same positive integer is invalid | Please fill in a valid email address | Please fill in a valid email address | Pass |
| 010 | Verify that user can not enter 0 in the quantity button  "I'm a product 12" | Enter a minimum amount of 1 | Enter a minimum amount of 1 | Pass |
| 011 | Verify section Artist Name is static if using multiple clicks on it | Section “Artist Name” is static if using multiple clicks on it | Section “Artist Name” is static if using multiple clicks on it | Pass |
| Ad-hoc testing | | | | | |
| 014 | Verify Email field does not allow to enter incomplete email | Please enter a valid email address | Please enter a valid email address | Pass |

Test Environment and Support (browsers):

● Laptop: Lenovo Inspiration 3793

● OS: Windows 11/64

● Google Chrome, version 114.0.5735.91 (Official Build) (64-bit)

The following test cases are covering automation tests (Selenium Python UnitTest):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Test requirements | Expected Test Result | Actual Test Results | Pass/Fail |
| Positive testing | | | | | |
| 002 | Verify login with valid username and password | User logged in the account | User logged in the account | Pass |
| 003 | Verify Chat functionality | User is able to chat, leave messages, insert icons (smiles etc.), attach files | User is able to chat, leave messages, insert icons (smiles etc.), attach files | Pass |
| 004 | Verify user is able to get notifications for upcoming show | User get notification | User did not get notification. | Fail |
| 005 | Verify user is able to proceed to Facebook and Twitter pages in "Artist name" block | User proceeds to Facebook and Twitter pages | User did not proceed to Facebook and Twitter pages. | Fail |
| Negative testing | | | | | |
| 007 | Verify that invalid email id and valid password is entered | Double check your email and try again. | Double check your email and try again. | Pass |
| 008 | Verify that invalid email id and invalid password is entered | Double check your email or password and try again. Passwords must be at least 4 characters long. Try again. | Double check your email or password and try again. Passwords must be at least 4 characters long. Try again. | Pass |
| 009 | Verify the completion of the Subscribe Form Email address using same positive integer is invalid | Please fill in a valid email address | Please fill in a valid email address | Pass |
| 010 | Verify that user can not enter 0 in the quantity button "I'm a product 12" | Enter a minimum amount of 1 | Enter a minimum amount of 1 | Pass |
| Ad-hoc testing | | | | | |
| 013 | Verify the Quantity box of "I'm a product3" doesn't exceed left in stock by entering massive number | Only 99999 lefts in stock | Only 99999 lefts in stock | Pass |

Test Environment and Support (browsers):

● Laptop: Lenovo Inspiration 3793

● OS: Windows 11/64

● Google Chrome, version 114.0.5735.91 (Official Build) (64-bit)

● Microsoft Edge, version 113.0.1774.57 (64-bit)

● Mozilla Firefox, version 113.0.2 (64-bit)

Testing Tools used:

|  |  |
| --- | --- |
| **Process** | **Tools** |
| Test case creation | PyCharm, Selenium |
| Test case tracking | PyCharm, Browsers |
| Test case execution | Automation |
| Test case management | PyCharm |
| Test reporting | HTML reports, Allure reports |

3.3 API test

API tests are executed to determine whether the APIs that are developed meet expectations when it comes to functionality, performance, reliability and security for the website.

Test Environment and Support (browsers):

● Laptop: Lenovo Inspiration 3793

● OS: Windows 11/64

● Google Chrome (DevTools)

The following test cases are covering API tests:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TC** | **Request / Test Action** | **Expected Test Results** | **Actual Test Results** | **Pass/ Fail** |
| 1 | POST / Add new address | New address is added  Status code is 200 | New address is added  Status code is 200 | Pass |
| 2 | POST / Make address by default | The address is set as default  Status code is 200 | The address is set as default  Status code is 200 | Pass |
| 3 | GET / Show all addresses | All addresses are being shown  Status code is 200 | All addresses are being shown  Status code is 200 | Pass |
| 4 | GET / Show default address | Default address is being shown  Status code is 200 | Default address is being shown  Status code is 200 | Pass |
| 5 | PATCH / Update existing address line 2 and Company name | Existing address and Company name are updated  Status code is 200 | Existing address and Company name are updated  Status code is 200 | Pass |
| 6 | GET / Show updated address | Updated address is being shown  Status code is 200 | Updated address is being shown  Status code is 200 | Pass |
| 7 | DEL / Delete existing address | Existing address is deleted  Status code is 200 | Existing address is deleted  Status code is 200 | Pass |
| 8 | GET / Check deleted address | The deleted address is not exist  Status code is 400 | The deleted address is not exist  Status code is 400 | Pass |

Testing Tools used:

|  |  |
| --- | --- |
| **Process** | **Tools** |
| Test case creation | Postman, Chrome DevTools |
| Test case tracking | Postman |
| Test case execution | Automation |
| Test case management | Postman |
| Test reporting | Newman |

3.4 Performance Automation 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, WebpageTest, BrowserStack-SpeedLab |
| Test case tracking | Google Lighthouse, GTMetrix, WebpageTest, BrowserStack-SpeedLab |
| Test case execution | Automation |
| Test case management |  |
| Test reporting | txt -, html -, pdf - files |

3.5 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, PentestTools |
| Test case tracking | Mozilla Observatory, ImmuniWeb, PentestTools |
| Test case execution | Automation |
| Test case management |  |
| Test reporting | txt -, html -, pdf - files |

**4. TESTS SCHEDULE AND ESTIMATED TIME**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sub-Task Name** | **Start** | **Est. time (hours)** | **Finish** |
| Manual Testing |  |  |  |
| Automation Testing |  |  |  |
| API Testing |  |  |  |
| Performance testing |  |  |  |
| Security testing |  |  |  |

Test Execution

QA -

Test phase -

Status -

APPROVALS:

|  |  |  |
| --- | --- | --- |
|  | Project Manager | QA Lead |
|  |  |  |
| Signature |  |  |