**EV Code Champion (Vision Script)**

**Test plan**

**Logo :**



**URL :**  [**https://66d586f1b075bddbbe64248b--illustrious-beijinho-778bd3.netlify.app/**](https://66d586f1b075bddbbe64248b--illustrious-beijinho-778bd3.netlify.app/)

**Table of Content :**

1. **Overview--------------------------------------------------------------------------------------**
2. **Introduction---------------------------------------------------------------------------------**
3. **Test Objective-------------------------------------------------------------------------------**
4. **Test Scope------------------------------------------------------------------------------------**
5. **Assumption------------------------------------------------------------------------------------**
6. **Test Items ------------------------------------------------------------------------------------**
7. **Test Strategy ----------------------------------------------------------------------------------**
8. **Test Environment ----------------------------------------------------------------------------**
9. **Test Schedule ---------------------------------------------------------------------------------**
10. **Test Cases-------------------------------------------------------------------------------------**
11. **Deliverables-----------------------------------------------------------------------------------**
12. **Risks & Mitigations-------------------------------------------------------------------------**
13. **Exit Criteria------------------------------------------------------------------------------------**

**Overview :**

**Project Name : EV Code Champion, Vision Script website**

Objective : Validate the functionality and performance of the EV Code Champion software testing approach. Ensure the proper functionality, usability, performance , and security of the EV Code Champion project hosted with the Vision Script website.

**1. Introduction**

**Scope**: This test plan focuses on validating the functionality of the EV Code Champion project within the Vision Script environment, identifying and resolving defects, and ensuring all components work as expected**.**

**Testing Approach:** Manual and automated testing methods will be used to verify different aspects of the system.

**2. Test Objectives**

Verify the EV Code Champion project meets functional and non-functional requirements.

Ensure seamless integration with the Vision Script website.

Validate data processing and interactions for the EV code champion logic.

**3. Test Scope**

In-Scope:

**Functional Testing:** Ensuring all core features of the EV Code Champion work, including code submission, validation, feedback, and leaderboards.

**Performance Testing:** Test how the project behaves under load, with multiple users or complex input.

**Security Testing:** Validate access control, data privacy, and protection against threats like SQL injections or XSS attacks.

**Usability Testing:** Ensure users have a seamless experience in terms of design, flow, and intuitiveness.

Out-of-Scope:

Testing of unrelated modules of the Vision Script platform.

**4. Assumptions**

The EV Code Champion project is already deployed or accessible on the Vision Script website.

Access to necessary test accounts or roles for different testing scenarios.

**5. Test Items**

**User Interface:** Check layout, navigation, buttons, forms, and user input validation.

**EV Code Logic:** Ensure that the logic for validating EV codes is working correctly.

Leaderboards: Ensure rankings update correctly based on user submissions.

**Data Validation:** Verify that user data (code submissions, profiles, etc.) is stored and processed correctly.

**6. Test Strategy**

**Types of Testing:**

**Unit Testing:** Testing the smallest units of the project (functions, methods).

**Integration Testing:** Ensure the EV Code Champion components integrate well with the Vision Script platform.

**Functional Testing:** Ensure all features (submissions, feedback, leaderboards, etc.) work as intended.

**Performance Testing:** Load testing to assess performance under heavy traffic.

**Security Testing:** Ensure security of the data and access levels.

**User Acceptance Testing (UAT):** Involve end users to validate usability and real-world scenarios.

**7. Test Environment**

**Test Environment:**

**Browser:** Chrome, Firefox, Safari, Edge (latest versions)

**Devices:** Desktop, mobile, and tablet

Operating System: Windows, macOS, iOS, Android

**Test Data:** Sample EV codes, user profiles, and inputs simulating various user behaviors.

**8. Test Schedule**

**Test Case Design:** Begin immediately, creating test cases for functional, performance, and security scenarios.

**Test Execution:** Upon completion of test case design, execute them over 2 weeks.

**Bug Reporting & Tracking:** Continuous process throughout the testing cycle, with defects being reported as soon as discovered

**9. Test Cases**

Some examples of test cases might include:

**Functional Test Cases:**

Verify user login and registration functionality.

Ensure valid EV code submissions return correct feedback.

Ensure invalid code submissions are properly handled.

**Usability Test Cases:**

Check that all buttons, links, and forms are responsive and accessible.

Verify that the leaderboard updates automatically after code submissions.

**Performance Test Cases:**

Simulate 1000 users submitting EV codes at the same time.

Measure response time of code validation.

**Security Test Cases:**

Check if an unauthenticated user can access restricted pages or features.

Test the platform against common web vulnerabilities like SQL injection or XSS.

**10. Deliverables**

Test case documentation

Test results summary

Defect reports

Final test report before production release

**11. Risks & Mitigations**

**Risk:** Delays in project completion may impact testing timelines.

**Mitigation:** Regular communication with development teams to track progress.

**Risk:** Limited test resources for performance and security testing.

**Mitigation:** Prioritize critical tests first and outsource testing if needed.

**12. Exit Criteria**

All critical test cases have passed.

No open defects classified as "high priority."

UAT has been completed successfully.

This test plan serves as a roadmap for testing the EV Code Champion project on the Vision Script website.

**Test Scenario :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test**  **Scenario id**  \_\_\_\_\_\_\_\_\_\_  TS\_001  \_\_\_\_\_\_\_\_\_\_\_  TS\_002  TS-003  TS-004  TS-005  \_\_\_\_\_\_\_\_\_\_\_  TS-006 | **Test Scenario**  **Description**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_  Verify that users can successfully  Register on the platform.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_  Verify the users can log into their  Accounts  Verify that user can select and submit coding  Challenges.  Verify that user can view the leaderboard and  Their ranking  Verify the users can manage their  Profile effectively.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Verify the users can logout  Securely. | **Preconditions**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Website is accessible  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  User account must  Be register  User must be log in  User has completed at least  One challenges.  User must be logged in  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  User must be logged in | **Test steps**  \_\_\_\_\_\_\_\_\_\_\_  1.Fill out the registration  Form with  Valid details  2.Submit the registration  Form  3. Attempt  To register  With an existing email  \_\_\_\_\_\_\_\_\_\_\_  1.Goto the  Login page  2.Enter valid credentials  3.Enter  Invalid  Credentials  1.Navigate  The challenges  Section  2.Select a  Challenges  Attempt  3.Write and  Submit a  Solution  1.Navigate the leaderboard  Section  2.verify the  Displayed  Ranking are  Correct  3. Click on a  User’s name to view their  Profile  1.Navigate  Profile setting  2. update  Profile  Information  3.change  Password  \_\_\_\_\_\_\_\_\_\_\_  1.Click on  Logout button.  2.Attempt  To access a  Restricted page.  3.Verify  Session  terminated | **Expected**  **Result**  \_\_\_\_\_\_\_\_\_\_  Registration  Page load  Successfully  All filled accept valid  Input.  Error message  Indicating the  Email is  Already in use  \_\_\_\_\_\_\_\_\_\_\_  Login page  Successfully  User is logged in  Successfully  Error message  Indicate invalid credentials  Challenged are displayed  Challenges  Details load  Correctly.  Submission  Confirmation  Appear with  Result.  Leaderboard  loads  successfully  Ranking reflect user’s  Scores accurately.  User’s profile  And challenges history load  Successfully  Profile management  Page loads  Changes save  Successfully  Password update successfully  \_\_\_\_\_\_\_\_\_\_\_  User logout  Successfully  Redirected to  The login page.  User can’t  Access the  Dashboard  Without  Logging. | **Test Environment**  \_\_\_\_\_\_\_\_\_\_\_  **1.Hardware**  **Requirement**  **.** Test machine  **.**Network  Infrastructure  \_\_\_\_\_\_\_\_\_\_\_\_  **2. Software**  **Requirement**  **.** Operating  System  (windows, mac OS,  Linux);  **3.** **Test data**  **.** User Accounts  . Challenges  Data  . Error  Handling  **4.** **Network**  **Configuration**  . Local Testing  . Staging  Environment  **5. Test**  **Management tools**  . Test case  Management  .Bug Tracking  \_\_\_\_\_\_\_\_\_\_\_\_  **6. Security and**  **Permissions** |

**Bug Report :**

**Date** : [ Insert Date of report]

Reported by :[ Alok Mishra]

**Project :** EV Code Champion, Vision Script Website

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Bug id**  **\_\_\_\_\_\_**  **BG\_001**  **\_\_\_\_\_\_**  **BG\_002**  **\_\_\_\_\_\_\_**  **BG\_003**  **\_\_\_\_\_\_**  **BG\_004**  **Bug id**  **BG\_005**  **BG\_006** | **Module**  **\_\_\_\_\_\_\_**  **Functional**  **\_\_\_\_\_\_\_\_**  **UI**  **Module**  **Performance** | **Bug**  **Description**  \_\_\_\_\_\_\_\_\_  1. Code editor does not  Display  Error  Message  Foe invalid  Syntax  \_\_\_\_\_\_\_\_\_  2.A link  In the  Document leads to  A 404  Error  \_\_\_\_\_\_\_\_\_  1.The code  Editor is  Misaligned on mobile  Device.  \_\_\_\_\_\_\_\_\_  2.Some  Sections  Have varying  Font size ,affecting  Readability.  **Description**  1.Running  Some simple code takes  Long time  \_\_\_\_\_\_\_\_  2.Documnet page load  Slow time | **Steps**  \_\_\_\_\_\_\_\_\_  Enter valid  Code and  Run it  \_\_\_\_\_\_\_\_  Click on  “Advanced  Error Handling”  **\_\_\_\_\_\_\_\_\_**  Open the  Editor on a  iPhone.  \_\_\_\_\_\_\_\_\_  Review the  Different  Sections  **Test steps**  Execute basic code  Snippets.  \_\_\_\_\_\_\_\_\_  Navigate to various  Documentation sections | **Expected**  **Output**  **\_\_\_\_\_\_\_\_\_**  An error  Message  Should be  Appear  \_\_\_\_\_\_\_\_\_  The correct  Documentation page  Should be load  \_\_\_\_\_\_\_\_\_\_  The layout  Should be  Adjust properly  \_\_\_\_\_\_\_\_\_  Consist the  Font size  Across all  Sections  **Expected**  **output**  Immediate  Response  \_\_\_\_\_\_\_\_\_  Page load  quickly | **Actual**  **Output**  **\_\_\_\_\_\_\_\_**  Error  Message  Should  Not  Appear  \_\_\_\_\_\_\_\_  Incorrect  Documentation  Page load  \_\_\_\_\_\_\_\_\_  Layout not adjust  properly  \_\_\_\_\_\_\_\_  Did not  Consist the  Font size  Across all  Sections  **Actual**  **output**  Take to  10 second above  \_\_\_\_\_\_\_\_  Very slow  Page load | **Priority/**  **Severity**  **\_\_\_\_\_\_\_\_\_**  High/  \_\_\_\_\_\_\_\_\_  High  \_\_\_\_\_\_\_\_\_\_  Medium  \_\_\_\_\_\_\_\_\_  Medium  **Priority**  Medium  \_\_\_\_\_\_\_\_\_  Medium | **Screenshot**  **Attachment**  **\_\_\_\_\_\_\_\_\_\_\_**  Screenshot  Attached in  Valid code  \_\_\_\_\_\_\_\_\_  Screenshot should be  Attached  Error handling  \_\_\_\_\_\_\_\_\_\_\_\_  Screenshot should be  Attached  In code editor  \_\_\_\_\_\_\_\_\_\_\_  Screenshot  Attached  In Review  Sections  **Comments**  Screenshot  Attached  In code  Running time  \_\_\_\_\_\_\_\_\_  Screenshot  Attached  In page load  section |

**Test Cases :**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TC\_ID**  **\_\_\_\_\_\_**  **TC\_001**  **TC\_002**  **TC\_003**  **TC\_004**  **TC\_005**  **TC\_006** | **Test case**  **Scenario**  **\_\_\_\_\_\_\_\_\_\_\_\_**  Testing string concatenation**.**  Testing the array sorting function.  Verify error handling for division by zero.  Testing variable assignment with different data types.  Testing basic arithmetic operations.  Testing built-in string functions. | **Test case**  **Description**  **\_\_\_\_\_\_\_\_\_\_\_**  Verify combining two strings using the + operator  Verify sorting an array in ascending order.  Test if Vision Script catches division by zero.  Verify that variables can hold various data types and that type conversion works as expected.  Verify that addition, subtraction, multiplication, and division work correctly.  Verify functions like string length, substring extraction, and case conversion. | **Pre –**  **Requisite**  **\_\_\_\_\_\_\_\_\_\_**  Vision Script is installed**.**  Vision Script array sort method.  Error handling function.  Vision Script environment is set up and supports multiple data types.  Arithmetic operators are implemented in Vision Script.  Vision Script provides string manipulation functions. | **Test**  **Steps**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Assign two strings: str1 = "Hello" and str2 = "World".  Concatenate: result = str1 + str2.  Create an array: arr = [5, 3, 9, 1].  Sort the array: arr.sort().  Write a division function: result = 10 / 0.  Capture error message.  Assign an integer: intVar = 10.  Assign a float: floatVar = 10.5.  Assign a boolean: boolVar = true.  Assign a string: strVar = "VisionScript".  Perform addition: sum = 5 + 3.  Perform subtraction: difference = 10 - 2.  Perform multiplication: product = 4 \* 5.  Perform division: quotient = 20 / 4.  Assign a string: text = "VisionScript".  Get the length: length = text.length().  Extract a substring: subText = text.substring(0, 6).  Convert to uppercase: upperText = text.toUpperCase(). | **Test data**  **\_\_\_\_\_\_\_\_\_\_\_\_**  "Hello", "World".  [5, 3, 9, 1].  10, 0.  Int Var = 10  Float Var = 10.5  Bool Var = true  strVar = "VisionScript"  Addition: 5 + 3  Subtraction: 10 - 2  Multiplication: 4 \* 5  Division: 20 / 4  "VisionScript" | **Expected**  **Output**  **\_\_\_\_\_\_\_\_\_\_\_**  "Hello World".  [1, 3, 5, 9].  Error message.  10  10.5  true  VisionScript  8  8  20  5  12  Vision  VISIONSCRIPT | **Actual**  **Output**  **\_\_\_\_\_\_\_** | **Status**  **\_\_\_\_\_\_** | **Comments** |

**Test Execution report :**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case id**  TC\_001  TC\_002  TC\_003 | **Test Scenario**  Homepage  Load  Feature check  Form submission | **Module**  Main page  Documentation  Contact form | **Test**  **Description**  Verify the  Homepage  Loads correct  Check if all  Feature are  Documented  Test form  Submission  functionality | **Tester**  John  Alok  Maya | **Execution**  **Date**  2024-10-3  2024-10-3  2024-10-3 | **Status**  Passed  Passed  passed | **Comment**  Loaded in 2 second  All feature  Documented  Form did not  submit |

**Test Summary report**

1. **Project information**

**Project Name :** EV CODE Champion , Vision script website

**Module :** login , logout, functional , UI, performance , main page , documentation .

**Testing period :** 2024-10-2 to 2024-10-4

**Prepared By :** [ Alok Mishra]

1. **Objective**

Verify the EV Code Champion project meets functional and non-functional requirements.

Ensure seamless integration with the Vision Script website.

Validate data processing and interactions for the EV code champion logic.

1. **Test Execution Summary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Total test**  **Cases** | **Executed** | **Passed** | **Filed** | **Blocked** | **In progress** |
| 20 | 17 | 15 | 2 | 1 | 4 |

1. **Defect summary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Defect ID** | **Severity** | **Priority** | **Module** | **Status** | **Remark** |
| BUG001 | High | Critical | Login | Open | Login button  Not functional  On safari browser |
| BUG001 | Medium | High | Form submission | Resolved | Delay form  Submission  Confirmation  screen |

1. **Environment Used for Testing**

**Browser :** Chrome, Firefox , safari

**Devices :** Windows, Android, iOS

**Test Tools :** Jira, Postman, Zephyer

1. **Risk & Mitigations**

Risk 1: Form submission failure

Mitigation: Investigate the issue in the form handler code. Ensure error handling is robust and validate fields on both client and server sides.

Risk 2: Slow page load times

Mitigation: Optimize images and scripts, implement lazy loading, and use content delivery networks (CDNs) to improve performance.

Risk 3: Incomplete or outdated documentation

Mitigation: Regularly update documentation and ensure it is in sync with the latest feature updates

1. **Conclusion**

The test execution on the Vision Script website confirmed that core functionalities such as page load times, feature documentation access, and navigation are performing as expected. While most tests passed, a few failures, such as form submission errors, were identified and require further attention. Overall, the website offers a solid user experience with a few areas for improvement. It is recommended to resolve the minor issues to ensure seamless functionality across all features. The site is close to being fully operational pending these fixes.