**Test plan for online store**

[**http://epicentrik.info/**](http://epicentrik.info/)

**Content**

1. Document Changes
2. Introduction
3. Test environment
4. Tested site functionality
5. Configurations
6. Testing process
7. Criteria for starting and ending testing
8. Work plan
9. Risks and restrictions
10. Test documentation
11. **Document Changes**

|  |  |  |
| --- | --- | --- |
| **Data** | **Creator** | **Kind of change** |
| 03/05/2019 | Karpenko O. | Creature |
| 04/05/2019 | Karpenko O. | Creature |
| 06/05/2019 | Rymarchuk O. | Editing |

**2 Introduction**

The purpose of this test plan is to describe the testing process of the online store “[**http://epicentrik.info/**](http://epicentrik.info/)”

***Test object***: [**http://epicentrik.info/**](http://epicentrik.info/).

***It is supposed to carry out:***

For individual fields:

* Positive application testing (correct steps, correct
* data).
* Negative testing (implies the introduction of incorrect
* data).

For the whole system:

* Functional testing;
* Cross-browser testing;
* Usability testing;
* User interface testing.

**3 Test environment**

|  |  |
| --- | --- |
| **Browsers** | Mozilla Firefox, Chrome, Safari, Opera |
| **OS** | Windows 8, Windows 10 |
| **Screen resolution** | 1366х768; 1280х800; 1280х1024; 1680х1050; 1929х1080 |
| **Mobile display with extension** | 480х800; 640x960; 768x1280; 1024x768; 1366x768 |

**4 Tested site functionality**

The following site functionality will be tested:

* product catalog - high priority;
* user basket - high priority;
* registration form - high priority;
* login;
* order submission form - high priority;
* user profile - low priority.

No stress and safety testing will be conducted due to the lack of necessary resources.

**5 Configurations**

|  |  |
| --- | --- |
| **Personal**  **a computer** | Windows 10 |
| **Site mapping to**  **mobile**  **devices** | HUAWEI GT3  Samsung Galaxy Star Plus GT-S7262  Iphone 7  IPad |

**6 Testing Process**

Testing Process:

For maximum coverage of sitescheck lists and test cases.

**7 Criteria for starting and ending testing**

Criteria for starting and ending testing:

Testing can be started if the following conditions are met:

1. The necessary documentation is ready and approved;

2. The tested functionality is completed and ready for transfer to testing.

Testing is completed if the following conditions are met:

1. All defects found are documented.

**8 Work plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time** | **Data start** | **Data finish** |
| **Drawing up a test plan and checklist** | 4 hours | 03.05.2019 | 03.05.2019 |
| **Adjustment test plan and**  **check list** | 1 hours | 04.05.2019 | 04.05.2019 |
| **Test execution** | 3 hours | 05.05.2019 | 05.05.2019 |
| **Writing a bug report** | 2 hours | 06.05.2019 | 06.05.2019 |

**9 Risks and restrictions**

Risks and restrictions: due to limited resources, ensure correct displaysite can only on devices that are listed in the table"Configurations."

**10 Test documentation**

Test documentation:

After testing, the availability of the following documents is planned:

* test plan;
* check list;
* bug report.

General Test Plan

* 1. Functionality testing
  2. Usability and Interface testing
  3. Compatibility testing
  4. Security testing
  5. Создать Traceability Matrix
  6. Написать Test Case Headers

**Comprehensive Testing Checklist for Testing Web and Desktop Applications:**

**Assumptions: Assuming that your application supports following functionality**

– Forms with various fields

– Child windows

– Application interacts with database

– Various search filter criteria and display results

– Image upload

– Send email functionality

– Data export functionality

**General Test Scenarios**

1. All mandatory fields should be validated and indicated by asterisk (\*) symbol

2. Validation error messages should be displayed properly at correct position

3. All error messages should be displayed in same CSS style (e.g. using red color)

4. Dropdown fields should have first entry as blank or text like ‘Select’

5. Delete functionality for any record on page should ask for confirmation

6. Select/deselect all records options should be provided if page supports record add/delete/update functionality

7. Amount values should be displayed with correct currency symbols

8. Default page sorting should be provided

9. All numeric values should be formatted properly

10. Input fields should be checked for max field value. Input values greater than specified max limit should not be accepted or stored in database

11. Check all input fields for special characters

12. Field labels should be standard e.g. field accepting user’s first name should be labeled properly as ‘First Name’

13. Check for timeout functionality. Timeout values should be configurable. Check application behavior after operation timeout

14. Check cookies used in an application

15. Check if downloadable files are pointing to correct file paths

16. All resource keys should be configurable in config files or database instead of hard coding

17. Standard conventions should be followed throughout for naming resource keys

18 Validate markup for all web pages (validate HTML and CSS for syntax errors) to make sure it is compliant with the standards

19. Application crash or unavailable pages should be redirected to error page

20 Check text on all pages for spelling and grammatical errors

21. Check numeric input fields with character input values. Proper validation message should appear

22. Check for negative numbers if allowed for numeric fields

23. Check amount fields with decimal number values

24. Check functionality of buttons available on all pages

25. User should not be able to submit page twice by pressing submit button in quick succession.

26. Divide by zero errors should be handled for any calculations

27. Input data with first and last position blank should be handled correctly

**GUI and Usability Test Scenarios**

1. All fields on page (e.g. text box, radio options, dropdown lists) should be aligned properly

2. Numeric values should be right justified unless specified otherwise

3. Enough space should be provided between field labels, columns, rows, error messages etc.

4. Scroll bar should be enabled only when necessary

5. Font size, style and color for headline, description text, labels, infield data, and grid info should be standard as specified in SRS

6. Description text box should be multi-line

7. Disabled fields should be grayed out and user should not be able to set focus on these fields

8. Upon click of any input text field, mouse arrow pointer should get changed to cursor

9. User should not be able to type in drop down select lists

10. Information filled by users should remain intact when there is error message on page submit. User should be able to submit the form again by correcting the errors

11. Check if proper field labels are used in error messages

12. Dropdown field values should be displayed in defined sort order

13. Tab and Shift+Tab order should work properly

14. Field specific and page level help messages should be available

15. Check if correct fields are highlighted in case of errors

16. Check if dropdown list options are readable and not truncated due to field size limit

17. All buttons on page should be accessible by keyboard shortcuts and user should be able to perform all operations using keyboard

18. Check all pages for broken images

19. Check all pages for broken links

20. All pages should have title

21. Confirmation messages should be displayed before performing any update or delete operation

22. Hour glass should be displayed when application is busy

23. Page text should be left justified

24. User should be able to select only one radio option and any combination for check boxes.

**Test Scenarios for Filter Criteria**

1. User should be able to filter results using all parameters on the page

2. Refine search functionality should load search page with all user selected search parameters

3. When there is at least one filter criteria is required to perform search operation, make sure proper error message is displayed when user submits the page without selecting any filter criteria.

4. When at least one filter criteria selection is not compulsory user should be able to submit page and default search criteria should get used to query results

5. Proper validation messages should be displayed for invalid values for filter criteria

**Test Scenarios for a Window**

1. Check if default window size is correct

2. Check if child window size is correct

3. Check if there is any field on page with default focus (in general, the focus should be set on first input field of the screen)

4. Check if child windows are getting closed on closing parent/opener window

5. If child window is opened, user should not be able to use or update any field on background or parent window

6. Check window minimize, maximize and close functionality

7. Check if window is re-sizable

8. Check scroll bar functionality for parent and child windows

9. Check cancel button functionality for child window

**Database Testing Test Scenarios**

1. Check if correct data is getting saved in database upon successful page submit

2. Check values for columns which are not accepting null values

3. Check for data integrity. Data should be stored in single or multiple tables based on design

4. Index names should be given as per the standards e.g. IND\_<Tablename>\_<ColumnName>

5. Tables should have primary key column

6. Table columns should have description information available (except for audit columns like created date, created by etc.)

7. For every database add/update operation log should be added

8. Required table indexes should be created

9. Check if data is committed to database only when the operation is successfully completed

10. Data should be rolled back in case of failed transactions

11. Database name should be given as per the application type i.e. test, UAT, sandbox, live (though this is not a standard it is helpful for database maintenance)

12. Database logical names should be given according to database name (again this is not standard but helpful for DB maintenance)

13. Stored procedures should not be named with prefix “sp\_”

14. Check is values for table audit columns (like createddate, createdby, updatedate, updatedby, isdeleted, deleteddate, deletedby etc.) are populated properly

15. Check if input data is not truncated while saving. Field length shown to user on page and in database schema should be same

16. Check numeric fields with minimum, maximum, and float values

17. Check numeric fields with negative values (for both acceptance and non-acceptance)

18. Check if radio button and dropdown list options are saved correctly in database

19. Check if database fields are designed with correct data type and data length

20. Check if all table constraints like Primary key, Foreign key etc. are implemented correctly

21. Test stored procedures and triggers with sample input data

22. Input field leading and trailing spaces should be truncated before committing data to database

23. Null values should not be allowed for Primary key column

**Test Scenarios for Image Upload Functionality**

(Also applicable for other file upload functionality)

1. Check for uploaded image path

2. Check image upload and change functionality

3. Check image upload functionality with image files of different extensions (e.g. JPEG, PNG, BMP etc.)

4. Check image upload functionality with images having space or any other allowed special character in file name

5. Check duplicate name image upload

6. Check image upload with image size greater than the max allowed size. Proper error message should be displayed.

7. Check image upload functionality with file types other than images (e.g. txt, doc, pdf, exe etc.). Proper error message should be displayed

8. Check if images of specified height and width (if defined) are accepted otherwise rejected

9. Image upload progress bar should appear for large size images

10. Check if cancel button functionality is working in between upload process

11. Check if file selection dialog shows only supported files listed

12. Check multiple images upload functionality

13. Check image quality after upload. Image quality should not be changed after upload

14. Check if user is able to use/view the uploaded images

**Test Scenarios for Sending Emails**

(Test cases for composing or validating emails are not included)

(Make sure to use dummy email addresses before executing email related tests)

1. Email template should use standard CSS for all emails

2. Email addresses should be validated before sending emails

3. Special characters in email body template should be handled properly

4. Language specific characters (e.g. Russian, Chinese or German language characters) should be handled properly in email body template

5. Email subject should not be blank

6. Placeholder fields used in email template should be replaced with actual values e.g. {Firstname} {Lastname} should be replaced with individuals first and last name properly for all recipients

7. If reports with dynamic values are included in email body, report data should be calculated correctly

8. Email sender name should not be blank

9. Emails should be checked in different email clients like Outlook, Gmail, Hotmail, Yahoo! mail etc.

10. Check send email functionality using TO, CC and BCC fields

11. Check plain text emails

12. Check HTML format emails

13. Check email header and footer for company logo, privacy policy and other links

14. Check emails with attachments

15. Check send email functionality to single, multiple or distribution list recipients

16. Check if reply to email address is correct

17. Check sending high volume of emails

**Performance Testing Test Scenarios**

1. Check if page load time is within acceptable range

2. Check page load on slow connections

3. Check response time for any action under light, normal, moderate and heavy load conditions

4. Check performance of database stored procedures and triggers

5. Check database query execution time

6. Check for load testing of application

7. Check for stress testing of application

8. Check CPU and memory usage under peak load condition

**Security Testing Test Scenarios**

1. Check for SQL injection attacks

2. Secure pages should use HTTPS protocol

3. Page crash should not reveal application or server info. Error page should be displayed for this

4. Escape special characters in input

5. Error messages should not reveal any sensitive information

6. All credentials should be transferred over an encrypted channel

7. Test password security and password policy enforcement

8. Check application logout functionality

9. Check for Brute Force Attacks

10. Cookie information should be stored in encrypted format only

11. Check session cookie duration and session termination after timeout or logout

11. Session tokens should be transmitted over secured channel

13. Password should not be stored in cookies

14. Test for Denial of Service attacks

15. Test for memory leakage

16. Test unauthorized application access by manipulating variable values in browser address bar

17. Test file extension handing so that exe files are not uploaded and executed on server

18. Sensitive fields like passwords and credit card information should not have auto complete enabled

19. File upload functionality should use file type restrictions and also anti-virus for scanning uploaded files

20. Check if directory listing is prohibited

21. Password and other sensitive fields should be masked while typing

22. Check if forgot password functionality is secured with features like temporary password expiry after specified hours and security question is asked before changing or requesting new password

23. Verify CAPTCHA functionality

24. Check if important events are logged in log files

25. Check if access privileges are implemented correctly