# **TEST CASES FOR AN ONLINE NEWSPAPER WEBSITE**

# **1. Test Plan**

***1.1. Introduction***

**Purpose:** Verify the website's login, search, article saving, and commenting functions when users access the website on laptops/computers.

**Testing scope:** login, search, articles saving, and commenting functions

**Objectives:** Ensure the website’s functions are operated smoothly and successfully, find bugs, and verify functionality.

**Testing methods:** manual testing

***1.2. Test Items***

- Online newspaper Website

- Relevant documentation: Test scenario, test cases, test data, bug report.

***1.3. Features to Be Tested***

- The Login function of the Website will be tested to ensure sure the user login successfully into the website

- The Search function of the Website will be tested to ensure the finding results appear when the user uses the search bar.

- The Article-saving function of the Website will be tested to ensure the user successfully saves an article on the website for later reading.

- The Commenting function of the Website will be tested to ensure the user can post comments on the website.

***1.4. Test Strategy***

**Testing Levels:** Unit testing level.

**Types of Testing:** Functional, usability, and performance.

**Approach:** All features will be tested by comparing the expected and actual results designed in the test scenario and test cases. The technique used is manual execution.

***1.5. Test Environment***

*Hardware Requirements:*

- Server Configuration:

* Processor: Intel Xeon 8 cores or equivalent
* RAM: 16 GB
* Disk Space: 500 GB SSD

- Devices: Desktop Windows

*Software Requirements:*

* Operating Systems: Windows 10
* Browsers: Chrome v115+
* Database: MySQL v8.0
* Web Servers: Apache HTTP Server v2.4

*Network Configuration:*

* Internet speed: Minimum 10 Mbps for testing page load times
* Test on both LAN and public network environments to simulate real user conditions.

*Testing Tools:*

* Bug Tracking Tool: Jira
* Test Case Management Tool: TestRail
* Browser Compatibility Testing: BrowserStack
* Performance Testing Tools: Google Lighthouse

***1.6. Test Schedule***

Created date: 15/6/2024

Executed date: 20/8/2024

***1.7. Entry and Exit Criteria***

Entry Criteria: The website is still working and environment-ready.

Exit Criteria: Achieving over 70% of test case execution and finding critical bugs.

***1.8. Risks and Contingencies***

- The wifi connection is poor during the testing duration.

- The hardware or software of the device is broken.

- The Website is shut down.

- The tester or executor is not available during the test duration,

- Any force majeure event occurs and delays the process.

***1.9. Responsibilities***

***Test Manager***

- Role: Oversees the entire testing process and ensures that testing is completed on time and meets quality standards.

- Responsibilities:

* Develop the test plan and strategy.
* Allocate resources and assign responsibilities to team members.
* Monitor test progress and track key metrics.
* Report test results and risks to stakeholders.
* Ensure adherence to test timelines and milestones.
* Facilitate communication between the testing team and other departments.

***Manual Tester***

- Role: Perform manual testing of the website to identify bugs and ensure that features are working as expected.

- Responsibilities:

* Design and execute test cases based on the requirements.
* Report, document, and track defects in the bug-tracking system.
* Perform regression testing to ensure that bug fixes do not introduce new issues.
* Conduct cross-browser and cross-device testing.
* Collaborate with developers and the test lead to ensure defect resolution.

***Developer***

- Role: Develop and maintain the news website, fixing bugs identified during testing.

- Responsibilities:

* Collaborate with testers to reproduce and understand defects.
* Fix bugs and ensure the code meets the requirements.
* Participate in code reviews to ensure quality.
* Perform unit testing and assist with integration testing before handing off to testers.

***Project Manager***

- Role: Ensures that the project runs smoothly, coordinating between various teams and ensuring alignment with business goals.

- Responsibilities:

* Communicate project scope, deadlines, and objectives to all team members.
* Monitor overall project progress and adjust timelines when needed.
* Ensure that project deliverables meet the expected quality.
* Facilitate meetings and reviews to discuss project updates and risks.

***Business Analyst***

- Role: Bridge the gap between the business stakeholders and the technical team, ensuring that requirements are communicated.

- Responsibilities:

* Define and document the business requirements and use cases.
* Work with testers to ensure that test cases align with business objectives.
* Clarify doubts and answer questions related to the business side of the project.
* Review test results to ensure that the website meets business goals.

***UI/UX Designer***

- Role: Responsible for the design and user experience of the website.

- Responsibilities:

* Collaborate with the testing team to review UI and usability issues.
* Ensure that design elements are implemented correctly and provide a seamless user experience.
* Provide design-related input for test cases.
* Review the layout and accessibility after testing.

***Stakeholders***

- Role: Ensure that the final product meets business goals and user expectations.

- Responsibilities:

* Review test results and provide feedback on the functionality and user experience.
* Prioritize features and defects based on business needs.
* Approve the final release of the website.

***1.10. Test Case Design***

Test case template

| Field | Description |
| --- | --- |
| Test Case ID | Unique identifier (e.g., TC001, TC002) |
| Test title | Describe the primary purpose or action being tested in the test case |
| Description | Detailed explanation of the test objective. |
| Preconditions | Any setup required before running the test. |
| Test Steps | Step-by-step instructions on how to execute the test case |
| Test Data | Input data used for the test. |
| Expected Result | What should happen if the system behaves correctly |
| Actual Result | What happens when the test case is executed (filled after execution) |
| Status | Pass/Fail (determined after execution) |
| Priority | Importance of the test case (High/Medium/Low) |
| Comments | Any additional notes or observations |

The criteria for defining test cases:

- Positive Test Cases: Verify that the system behaves as expected with valid input and under normal conditions.

- Negative Test Cases: Test how the system handles invalid inputs or unexpected actions.

- Boundary Test Cases: Test the behavior of the system at the edges of acceptable input ranges.

- Edge Test Cases: Test cases that examine the behavior of the system under extreme conditions or unusual circumstances.

***1.11. Defect Management***

*- Identifying Defects*

+ Process: While executing test cases, the tester may encounter unexpected behavior or functionality that deviates from the expected result.

+ Criteria for Defect Identification:

* The actual result does not match the expected result in a test case.
* A feature behaves incorrectly or inconsistently across browsers or devices.
* Performance issues or usability problems.
* Security vulnerabilities or accessibility problems.

*- Logging Defects*

+ Process: Once a defect is identified, it needs to be logged in a defect tracking tool for communication and resolution.

+ Tool Used for Defect Tracking: Jira, Bugzilla, or Trello.

+ Details to Include in a Defect Report:

* Defect ID: A unique identifier automatically generated by the tool.
* Summary: A brief description of the defect.
* Description: A detailed explanation of the defect, including steps to reproduce it, the expected result, and the actual result.
* Severity/Priority: Classify the defect based on its impact (e.g., Critical, High, Medium, Low).
* Environment: Specify the test environment where the defect was found.
* Attachments: Include screenshots, videos, or logs that demonstrate the defect.
* Assigned to: The developer or team responsible for fixing the defect.

*Tracking Defects*

- Process: Once logged, the defect will move through various statuses until it is resolved. Tracking ensures the team is aware of the defect's progress.

- Defect Life Cycle:

* New: The defect is logged and is under initial review.
* Assigned: The defect is assigned to a developer for fixing.
* In Progress: The developer is actively working on resolving the defect.
* Fixed: The developer claims the defect is fixed.
* Re-Test: The tester re-executes the test case to verify the fix.
* Closed: The defect is confirmed as resolved, and no further action is needed.
* Reopened: If the issue persists after testing, the defect is reopened for further investigation.

*Resolving Defects*

- Process:

* Once the developer has fixed the defect, it is moved to the "Fixed" status.
* The manual tester will then re-execute the test case that originally identified the defect to verify the fix.
* If the defect no longer occurs, the tester moves it to "Closed" status.
* If the issue still exists, the tester will reopen the defect and provide additional details for further analysis.

# **2. Test Scenarios**

Created by: Ngoc Tram

Created date: 15/6/2024

Executed by: Minh Quang

Executed date: 20/8/2024

***2.1. Checking the login function*** Scenario ID: SE\_01

Test case

Created by: Ngoc Tram

Created date: 15/6/2024

Executed by: Minh Quang

Executed date: 20/8/2024

- TE\_01: Enter the correct email and password

- TE\_02: Enter the correct email, but false password

- TE\_03: Enter the correct password, but false email

- TE\_04: Enter the false email and password

- TE\_05: Leave the email field blank

- TE\_06: Leave the password field blank

***2.2. Checking the search function*** Scenario ID: SE\_02

Test case

Created by: Ngoc Tram

Created date: 15/6/2024

Executed by: Minh Quang

Executed date: 20/8/2024

- TE\_07: Search for a meaningful keyword

- TE\_08: Search for a special character

- TE\_09: Search for a number

- TE\_10: Leave the search bar blank

***2.3. Checking the article saving function*** Scenario ID: SE\_03

Test case

Created by: Ngoc Tram

Created date: 15/6/2024

Executed by: Minh Quang

Executed date: 20/8/2024

- TE\_11: Save an article after logging into the account.

- TE\_12: Save an article without logging into the account.

***2.4. Checking the commenting function*** Scenario ID: SE\_04

Test case

Created by: Ngoc Tram

Created date: 15/6/2024

Executed by: Minh Quang

Executed date: 20/8/2024

- TE\_13: Post a comment after logging into the account.

- TE\_14: Post a comment without logging into the account.

- TE\_15: Leave the comment box blank