

Website Gadget House

Test-plan

Version 1.0

Website GadgetHouse	Version 1.0
Test Plan	Date: 01.06.2024

Changes

Date	Version	Type	Author
01.06.2024	1.0	Creation	Vira Dovhalska
22.12.2024	2.0	Editing	Vira Dovhalska
10.01.2025	2.1	Planned extension of functionality with API	Vira Dovhalska

Contents

1. Introduction	4
Objective	4
Input data	4
Testing objective	4
2. Testing Conditions	5
3. Testing Strategy	8
Functional Testing	9
Cross-browser and Mobile Testing	12
Regression Testing and verification of resolved defects	13
Design Testing	14
4. Work plan	14
5. Final results	15
Conclusion	15

1. Introduction

Objective

The purpose of creating this test plan is to describe the process of testing the Gadget House website (full address: <https://gadget-house-tc.netlify.app>). The document provides an overview of the planned testing activities for the project.

Input data

The Gadget House website allows users to obtain detailed information about the company, view new products, and place orders for company products (smartphones, laptops, game consoles and other gadgets).

Testing objective

The purpose of testing the Gadget House website is to verify the correct functioning of all its features across different browser versions under typical usage scenarios. Approximately 20% of the testing time will be allocated to testing non-typical usage scenarios or those that potentially trigger operational errors.

The testing process will result in the following materials:

- **QA Report on Overall Status:** This will provide developers and product manager with insights into the correct functionality of the website across different browsers.
- **Test Results Report:** This will cover the current testing coverage, typical use cases, and browsers.
- **Documented Bugs in Trello:** All identified issues will be reported in Trello.

Testing will be conducted manually using "ad-hoc testing" from the perspective of the website's end-user.

KPIs (Key Performance Indicators) :

- **Test coverage percentage:** Aim to achieve 90% coverage of functional requirements.
- **Defect Fixing Time:** Critical defects should be resolved within 2 business days on average.
- **Test case success rate:** 80% of test cases must pass on the first attempt.

2. Testing Conditions

The testing of the website will ensure it meets the needs of users for interacting with e-commerce products, providing a seamless and intuitive experience for browsing, detailed product descriptions, and order processing. The following conditions are considered critical for thorough testing:

1. User Interface and User Experience (UI/UX)

- **Responsiveness:** The website should be fully responsive and function correctly across various devices and screen sizes (e.g., desktop, tablet, mobile).
- **Navigation:** All navigation elements, including menus, buttons, and links, should be intuitive and accessible.
- **Consistency:** Design elements such as fonts, colors, and layouts should be consistent across all pages.
- **Accessibility:** The website should comply with accessibility standards (e.g., WCAG 2.1), ensuring it is usable by people with disabilities.

2. Product Interaction

- **Browsing:** Users should be able to browse product categories and subcategories smoothly, with accurate filtering and sorting options.
- **Product Details:** Detailed product descriptions, specifications, and high-quality images should be clearly visible and provide all necessary information.

3. Search Functionality

- **Accuracy:** The search function should return relevant results based on keywords, product names, and descriptions.
- **Speed:** Search results should be displayed quickly without significant delay.

4. Photo Gallery

- **Display:** The photo gallery should display images without distortion and allow for zooming in on details.
- **Navigation:** Users should be able to easily navigate between different images for the same product.

5. Shopping Cart and Order Processing

- **Cart Management:** Users should be able to add, remove, and update quantities of products in the shopping cart effortlessly.
- **Checkout Process:** The checkout process should be streamlined and secure, guiding users through order placement, payment, and confirmation steps efficiently.
- **Payment Methods:** Ensure that the shopping cart and order process simulate the selection of payment methods without actual payment processing.
- **Order Tracking:** After placing an order, users should be able to track their order status through their account.

6. Performance

- **Loading Times:** Pages should load within an acceptable time frame to ensure a positive user experience.
- **Scalability:** The website should handle a high volume of users and transactions without performance degradation.

7. Security

- **Data Protection:** User data, including personal information, should be securely handled and encrypted.
- **Authentication:** Login and account management processes should be secure and protect against unauthorized access.
- **Vulnerabilities:** The site should be tested for common vulnerabilities such as SQL injection, XSS, etc.

8. Compatibility

- **Browsers:** The website should be compatible with major browsers (Chrome, Firefox, Safari, Edge) and their latest versions.
- **Operating Systems:** It should function correctly on major operating systems (Windows, macOS, iOS, Android).

9. Error Handling and Recovery

- **Graceful Degradation:** In case of errors, the website should display user-friendly messages and provide recovery options.
- **Session Management:** Proper handling of user sessions, especially in scenarios such as timeouts or interruptions during order placement.

10. Content Management

- **Updates:** The CMS (Content Management System) should allow for easy updating of product details, images, and other content without impacting the user experience.
- **Localization:** If the website will support multiple languages, content should be accurately translated and displayed.

11. Testing Environment

- **Development Environment:** Testing will be conducted in a development environment that mirrors the production environment as closely as possible.
- **Data:** Use of realistic but non-sensitive test data to replicate actual user scenarios.

3. Testing Strategy

The plan below is formal, as a detailed plan requires an understanding of the current project status. The initial functional tests will help refine the test plan. The first run will provide an accurate view of system stability, and a clear set of tests will be determined for each configuration.

This approach will produce a detailed report on the tested product and focus on critical areas. Weekly reports will be provided to the client, detailing the progress of testing, discovered defects, and suggestions for product and design improvements. All identified defects will be logged in Trello as individual tickets for further correction.

Due to the lack of specifications and limited resources for test formalization, ad-hoc testing will be used during the Gadget House website testing. The testing process is planned in five stages:

1. **Technical Analysis and Test Plan Preparation:** Partial run of functional tests.
2. **Detailed Functional Testing:** Identification and description of defects.
3. **Cross-Browser and Mobile Testing:** Description of identified defects.
4. **Bug Verification and Regression Testing:** Check the fixed issues by developers.
5. **Design Testing:** Description of identified defects.

This allows for maximum testing depth, enabling the project developers to correct defects early.

Approved OS for Testing:

- Windows 10 x64

Approved Browser Versions for Testing:

- Google Chrome 120.0.6422.141
- Mozilla Firefox 121.0
- Microsoft Edge 120.0.2210.91

Security and Stress Testing

Security Testing:

1. **Authentication Testing:** Verify that only authorized users can access the system. Test login functionality with valid and invalid credentials to ensure proper authentication.
2. **Authorization Testing:** Check that users have appropriate permissions based on their roles. Attempt to access restricted areas/functions without proper authorization to confirm

access controls.

3. **Input Validation Testing:** Test all input fields (e.g., forms, search bars) to ensure they accept only valid data. Attempt to input malicious or invalid data (e.g., SQL injection attempts) to verify robustness against attacks (e.g., ' OR 1=1 --). Test for XSS vulnerabilities by entering JavaScript code snippets in fields (e.g., '<script>alert('XSS')</script>'). Verify email duplication error when trying to register with an already used address.
4. **Session Management Testing:** Validate that session tokens are securely managed throughout the user's session. Test for session fixation, expiration, and regeneration upon login/logout.
5. **Data Security Testing:** Ensure that sensitive data is stored securely. Check for encryption during transmission (e.g., HTTPS) and storage (e.g., hashed passwords).
6. **Error Handling Testing:** Test how the system handles various error conditions gracefully. Ensure that error messages do not reveal sensitive information and are informative yet secure.

Stress testing will be conducted upon agreement of necessity.

Types of Testing

Functional Testing

Objective: Identify functional errors, inconsistencies in technical support, and user expectations through standard and non-trivial test scenarios.

Process Description:

- Registration/Authorization
 - User Registration
 - User Authorization
 - Anonymous User Login
 - Password Recovery
 - Account Editing
- Personal Account
 - Editing Personal Data (Updating contact information, delivery address, and personal settings)
 - User Account Deletion (Request for account deletion, confirmation via email)
 - Order Status Display (Viewing active and completed orders)

- User Logout from Personal Account
- Search
 - Keyword Entry for Search
 - Viewing Search Results and Their Correspondence to Input Criteria
 - Handling Search Queries in Different Languages
 - Displaying Correct Results for Incorrect Queries
- Filtering and Sorting
 - Checking the Correct Operation of Category Filters
 - Sorting Products by Various Parameters (price, rating)
- Order Processing
 - Adding Products to Favorites
 - Adding Products to Cart and Processing Orders
 - Correct Filling of the Order Form
 - Displaying Information About Saved Data
- Catalog and Product Page
 - Correct Display of Products in Groups
 - Quick Add to Cart and Favorites or Direct Purchase Buttons
 - Product Availability Functionality
 - Correct Transition to Specific Product Page and Return to Previous Page
 - Special Offers Functionality Related to the Selected Product
 - Additional Parameters Functionality (color selection, etc.)
- Cart
 - Add/Remove Product Functionality
 - Correct Display of Products in the Cart
 - Saving Cart Products on Browser Crash
 - Product Information Verification (name, price, quantity) before Purchase Confirmation
 - Change Product Quantity Functionality
 - Clear All Products Functionality
 - Total Price Calculation Functionality
 - Purchase Processing Functionality
 - User Product Return Functionality (Procedure, Feedback, Email Response)
- Footer
 - Links to Main Sections of the Site
 - Correct Link Transitions to Social Networks
 - Privacy Policy Functionality
 - Address Link Verification (Correct Map Display, Return to Site)

- Sending Email Through the Site Functionality
- **Input Validation Testing**
 1. Validation of text fields:
 - **Registration and authorization forms:**
 - Email: Must follow the standard format (e.g., user@example.com).
 - Password: Minimum 8 characters, max 24 characters, at least : one number, one lowercase and one uppercase Latin character and one special character.
 - Phone number: Format “xxxxxxxxxx” or “+xxxxxxxxx”.
 - **Search field:**
 - Minimum length: 2 characters.
 - Maximum length: 50 characters.
 - Validation rules: Alphanumeric characters are allowed (A-Z, a-z, 0-9). Special characters are accepted.
 - Prohibited input: Empty input or whitespace-only input; Attempts to inject scripts, HTML tags, or database queries (e.g. DROP TABLE, <script>); Unicode characters not related to alphanumeric input (e.g., emojis, non-standard symbols like ♔); Database-specific special characters, such as: asterisk(*), percent(%), single/double quotes(',"), backtick(`), angle brackets(<,>) , ampersand(&), caret(^), pipe (|), and backslash(\).
 - Error message: “*You may have entered an incorrect query. Check the spelling. Try to use only keywords.*”
 - **Feedback field:**
 - Maximum length: 500 characters.
 - Allowed characters: text, numbers and punctuation.
 2. Edge case testing:
 - Large input size: Verify truncation or handling of strings longer than the maximum allowed, and confirm that the user is notified of any transaction.

- Special characters: Ensure no injection vulnerabilities by blocking restricted inputs.
- Prohibited input: ensure fields reject any unsupported formats, encoding, or symbols(e.g., SQL commands, Unicode emojis).
- Missing required fields: Validate that submission is blocked if mandatory fields are empty.

Cross-Browser Testing

Objective: Ensure the project works correctly and maintains its design integrity across different browsers.

- Approved browsers: Google Chrome, Mozilla Firefox, Microsoft Edge
- Approved OS: Windows 10x64
- Mobile devices: iPhone 11 (Safari Browser, iOS 17), Samsung Galaxy A50(Samsung Internet Browser, Android 12)

Cross-Browser Testing Steps

1. Setup Testing Environment:
 - Install and configure all approved browser versions on Windows 10 x64.
2. Visual Inspection:
 - Check the visual appearance and layout of the website in each browser.
 - Look for discrepancies in fonts, colors, spacing, and element alignment.
3. Functional Testing:
 - Perform key user actions such as navigation, form submissions, and interaction with dynamic elements.
 - Ensure that all features function as expected.
4. Responsive Design Testing:
 - Resize the browser window to various dimensions and verify that the website adapts correctly.
 - Test on different devices to ensure proper functionality and layout.
5. Performance Testing:
 - Measure the load times and responsiveness of the website in different browsers.
 - Identify and log any performance issues.
6. Accessibility Testing:
 - Check that the website is accessible and keyboard navigation support.

Regression Testing and Verification of Resolved Defects

Objective: Identify any new defects or issues that might arise from changes in the source code or the introduction of new features. Confirm that the existing functionality continues to work correctly after changes.

During regression testing, the following tests will be conducted:

- Core Functionality
 - User Registration and Authorization
 - Adding Products to the Cart and Order Processing
 - User Personal Account
- Cart Functionality
 - Adding and Removing Products from the Cart
 - Calculating the Total Price of Products in the Cart
 - Correct Application of Promotions and Discounts
- Search and Navigation Testing
- Footer and Links Testing

Regression Testing Steps

1. Identify Test Cases:
 - Select and prioritize test cases that cover critical functionality and areas affected by recent changes.
2. Execute Tests:
 - Perform the selected test cases to verify that existing functionality remains unaffected by code changes.
3. Log Defects:
 - Document any new issues discovered during regression testing in Trello for further action.
4. Retest Fixed Issues:
 - Once defects are resolved, retest the affected functionality to ensure the issues are fixed.
5. Verify Overall Stability:
 - Ensure the website remains stable and functional across all tested scenarios after implementing changes.

Design Testing

Objective: Ensure the product design matches the specification mockups and displays correctly (adaptability for different screen sizes).

Process Description:

- Registration Form
 - Ensure alignment and functionality match the design specs.
- Personal Account
 - Verify the design and layout of personal account features.
- Website Pages
 - Confirm that all website pages adhere to the design guidelines.
- Adaptability for Different Screen Sizes
 - Check that the display is not disrupted by various browser configurations.
- Color Palette and Styles Testing
 - Ensure colors and styles conform to design specifications.
- Image Testing
 - Verify that images are displayed correctly across different browsers and screen sizes.
- Fonts and Readability Testing
 - Confirm that fonts are rendered properly and text is readable on different devices and browsers.

4. Work Plan

Task	Work Load	Start Date	End Date
Creating the Test Plan	14 days	1/06/24	14/06/24
Performing the Testing			
Analyzing the Testing			
Summarizing Result			

5. Final Results

1. Overview of Testing Outcomes

- **Objective:** Summarize the overall objectives of the testing process, including the primary goals and areas of focus.
- **Scope:** Recap the scope of testing, including what was tested and any areas not covered.

2. Testing Summary

- **Test Cases Executed:** Provide a summary of the total number of test cases executed, including the breakdown of passed, failed, and blocked test cases.
- **Defects Identified:** Summarize the total number of defects identified during testing, categorized by severity (e.g., critical, major, minor) and status (e.g., open, resolved, closed).
- **Defect Density:** Calculate and include defect density, which is the number of defects per unit size of the software.

3. Analysis of Results

- **Success Rate:** Indicate the percentage of test cases that passed versus the total number executed.
- **Failure Analysis:** Provide an analysis of the reasons for test failures, including trends or patterns observed in the defects.
- **Coverage Analysis:** Assess the test coverage achieved, indicating the extent to which the testing addressed the requirements and specifications.

4. Recommendations

- **Improvements:** Provide actionable recommendations for improvements based on the identified defects and test results. These could include changes to the code, enhancements to the user interface, or modifications to the workflow.
- **Future Testing:** Suggest areas for future testing or additional test cases that should be considered to further improve the product's quality and reliability.

5. Lessons Learned

- **Challenges Faced:** Document any challenges or issues encountered during the testing process and how they were addressed.
- **Best Practices:** Highlight any best practices that emerged from the testing process that could be useful for future projects or phases.

6. Conclusion

- **Summary:** Conclude with a summary of the testing outcomes, the quality of the product as evaluated through the testing, and the readiness of the product for release or further development.
- **Release Readiness:** Indicate whether the product meets the criteria for release, and if not, what actions need to be taken before it can be considered ready.