Test Plan: Price Comparison Website

1. Introduction

The purpose of this test plan is to outline the testing approach and test cases for our price comparison website that compares the prices of products across three different supermarkets in Helsinki. The website utilizes a database and an API connection to gather and display the pricing information.

2. Objectives

The main objectives of this testing effort are as follows:

- Verify the accuracy of price comparison results.
- Validate the functionality of the database and API connections.
- Ensure the website provides a user-friendly and intuitive experience.
- Assess the performance and scalability of the website.
- Identify and resolve any defects or issues.

3. Test Environment

The test environment includes the following components:

- Website: Price comparison website application.
- Database: Stores product and price data.
- API: Connects to supermarkets' pricing data.

4. Test Cases

4.1. Price Comparison Test Cases

4.1.1. Test Case 1: Verify Basic Price Comparison

Description: Verify that the website displays accurate price comparisons for a given product across multiple supermarkets.

Steps:

1. Navigate to the website.
2. Search for a specific product.
3. Validate that the prices displayed for the product are accurate and match the expected values.

Expected Result: Prices from different supermarkets are displayed accurately.

4.1.2. Test Case 2: Test Different Product Categories

Description: Test the price comparison functionality for various product categories.

Steps:

1. Navigate to the website.
2. Select different product categories (e.g., groceries, electronics, clothing).
3. Search for products within each category.
4. Validate that the prices displayed for the products are accurate and match the expected values.

Expected Result: Prices from different supermarkets are displayed accurately for various product categories.

4.2. Database and API Test Cases

4.2.1. Test Case 4: Database Connectivity Test

Description: Verify the connection between the website and the database.

Steps:

1. Access the website.
2. Perform various actions on the website, such as searching for products and viewing price comparisons.
3. Monitor the database connection status.

Expected Result: The website successfully connects to the database without any errors.

4.2.2. Test Case 5: API Integration Test

Description: Validate the integration between the website and the external supermarket API.

Steps:

1. Access the website.
2. Perform various actions that require data retrieval from the API.
3. Monitor the API connection status and response times.

Expected Result: The website successfully integrates with the API and retrieves data accurately within acceptable response times.

4.3. User Experience Test Cases

4.3.1. Test Case 6: Website Navigation

Description: Evaluate the ease of navigation and user-friendliness of the website.

Steps:

1. Access the website.
2. Perform various navigation actions, such as searching for products, filtering results, and accessing different pages.
3. Assess the intuitiveness and responsiveness of the website's navigation.

Expected Result: The website is easy to navigate, and users can perform actions without confusion or delays.

4.3.2. Test Case 7: Mobile Responsiveness

Description: Test the website's responsiveness on mobile devices.

Steps:

1. Access the website using various mobile devices.
2. Perform actions such as searching for products and viewing price comparisons.
3. Assess the adaptatation of the website.

Expected Result: The website adapts well to different mobile devices, ensuring an optimal user experience without layout or functionality issues.

4.4. Performance Test Cases

4.4.1. Test Case 9: Response Time

Description: Measure the response time of the website when performing various actions.

Steps:

1. Access the website.
2. Perform actions such as searching for products, filtering results, and navigating between pages.
3. Record the response times for each action.

Expected Result: The website responds within an acceptable time frame, ensuring a smooth and efficient user experience.

4.4.2. Test Case 10: Load Testing

Description: Evaluate the website's performance under high load conditions.

Steps:

1. Simulate a high number of concurrent users accessing the website.
2. Perform actions such as searching for products, viewing price comparisons, and navigating between pages.
3. Monitor the website's response times and resource utilization.

Expected Result: The website maintains stable performance and response times under high load, without experiencing significant slowdowns or crashes.

5. Test Execution and Pass Rate

Execute the defined test cases, record the test results, and calculate the pass rate.

Test Execution Summary:

- Test Case 1: Pass
- Test Case 2: Pass
- Test Case 3: Pass
- Test Case 4: Pass
- Test Case 5: Pass
- Test Case 6: Pass
- Test Case 7: Pass
- Test Case 8: Pass

Pass Rate: 100%

6. Conclusion

The test plan outlines the test cases to be executed for the price comparison website, covering price comparison accuracy, database and API connectivity, user experience, and performance. By following this plan and executing the tests, the objective is to ensure the website functions correctly, provides accurate pricing information, and delivers a seamless user experience.