Test Plan Of

Buyoniasoft Buddy-Tutor

An Online Based Student-Tutor Platform

Table of Content

1. Introduction

The purpose of this test plan is to define a comprehensive and structured approach to testing the Buddy Tutor website developed by Buyoniasoft. The goal is to ensure that the website is functioning as intended and meets the requirements specified in the project scope. This test plan outlines the scope of the testing, the testing approach, the test scenarios, and the test cases.

2. Test Scope

The scope of testing includes the following:

- Functional testing to ensure that all website features and functionalities are working as expected.
- User interface testing to ensure that the website is user-friendly and easy to navigate.
- Compatibility testing to ensure that the website works correctly on different browsers and devices.
- Performance testing to ensure that the website is responsive and performs well under various traffic loads.
- Security testing to ensure that the website is secure and protected against common attacks.
- API testing to ensure that the API functions correctly, securely, and efficiently and that it meets the expected requirements and standards.
- Test Automation includes the functionality of the various features, data integrity and accuracy of the information, the performance of the website in different browsers and operating systems, and any other user-facing features.

3. Test Strategy

3.1 Testing Goals: The testing goals for the website, include functional requirements, performance requirements, security requirements, usability requirements, and compatibility requirements. Prioritize the testing goals based on their criticality and impact on the user experience.

3.2 Test Scenarios: Test scenarios based on our testing goals, including test cases for navigation, user authentication, user registration, profile management, payment processing, course enrollment, course completion, and user feedback as follows:

- 3.2.1 Functional Requirements

1. A user login system for students, tutors, and parents.
2. The ability for users to search for tutors based on criteria such as subject matter, location, and availability.
3. The ability for users to book tutoring sessions and view upcoming sessions.
4. The ability for tutors to list their services set their availability and manages the sessions they have booked.
5. The ability for users to rate and review tutors.
6. The ability for users to pay for sessions.

- 3.2.2 Performance Requirements

1. The application should run smoothly and quickly on all modern web browsers.
2. The application should respond quickly to user interactions and requests.
3. The application should have minimal downtime.

- 3.2.3 Security Requirements

1. Configuration and Deployment Management.
2. Test File Extensions Handling for Sensitive Information.
3. HTTP method.
4. HTTP Strict Transport Security.
5. File Permission.
6. Identity Management.
7. Authentication Schema.
8. Strong password policy and CAPTCHA.
9. Authorization.
10. Session Management.

11. Input Validation.
12. Cryptography and end-to-end encryption.
13. Client-side protection.

❖ 3.2.4 Usability Requirements

1. The application should be intuitive and easy to use.
2. The application should have a clear and concise user interface.
3. The application should provide helpful feedback to users when they make mistakes.

❖ 3.2.5 Compatibility Requirements

1. The application should be compatible with all modern web browsers.
2. The application should be compatible with mobile devices.
3. The application should be compatible with all major operating systems.

3.3 Testing Tools: The following testing tools will be used to test this application:

- 3.3.1 K6: It is a performance testing tool used to check the performance of the website. It can be used to measure response time, throughput, and many other performance metrics.
- 3.3.2 Cypress: It is an open-source testing tool that is used to test web applications for both front-end and back-end. It is designed to help developers and testers create automated tests that are easy to write, maintain, and debug. Cypress works by running tests directly in the browser, allowing developers to quickly and easily create tests that are fast and reliable.
- 3.3.3 Postman: It is a powerful tool that allows users to test, develop, and document APIs (Application Programming Interfaces). APIs are used to allow two different applications to communicate with each other. Postman helps developers

and testers easily test, debug, and document APIs, without the need to write any code.

- 3.3.4 Burp Suite: It is an integrated platform for performing security testing of web applications. It provides a comprehensive set of manual and automated tools for attacking web applications and services, including scanning for vulnerabilities, fuzzing, decoding, and analyzing web traffic, and more.

- 3.3.5 Nmap: It is a powerful network mapping and port scanning tool used by network professionals to identify and troubleshoot network devices and services. Nmap works by sending a series of packets to a target host and analyzing the responses. It can be used to detect open ports, identify services running on those ports, detect security vulnerabilities, fingerprint operating systems, and much more.

- 3.3.6 Nikto: It is a free, open-source web application scanner tool used for identifying potential vulnerabilities in web servers. It is a part of the Kali Linux distribution and is used to detect misconfiguration, outdated server software, CGI-based vulnerabilities, and other potential security issues. It can also detect default installation files, backup files, and other common security problems.

3.4 Test Data: This is a web-based tutoring platform that includes student tutor and parent profiles, lesson plans, payment details, etc. The test data will include valid and invalid user credentials, payment information, course details, and feedback data.

3.5 Test Execution: Executing all the tests based on scenarios using the selected testing tools and test data. Recording the test results in the Taiga.

Taiga is an open-source project management tool designed to help teams manage projects, tasks, and resources more efficiently. Document any defects or issues encountered during testing and track them until resolution.

3.6 Test Results Analysis: Analyze the test results to identify trends, patterns, and areas for improvement. Prioritize the defects based on their criticality and impact on the user experience. Provide feedback to the development team on the areas that need improvement.

3.7 Retest: Retest the defects that have been resolved by the development team to ensure that they have been fixed correctly. Verify that the website meets the testing goals and that the user experience is satisfactory.

4. Test Environment

- 4.1 Platform
    - Smartphone.
    - Tablet.
    - PC.

- 4.2 Operating System
    - Android.
    - IOS.
    - Windows.
    - Linux.
    - Mac.

- 4.3 Browser
    - Google Chrome.
    - Mozilla Firefox.
    - Opera.
    - Safari.
    - UC.

5. Test Methodologies

5.1 System Testing: In here, this completed and integrated application is tested to verify that all components of the system work as expected. The purpose of system testing is to validate that the software system meets its requirements and behaves as expected.

5.2 Acceptance Testing: In here, this application is tested in its entirety to verify that it meets the requirements specified by the customer. The purpose of acceptance testing is to confirm that the system meets the customer's requirements and can be used in a production environment.

6. Test Deliverables

6.1 Test Cases: A comprehensive suite of test cases, covering all functional and non-functional requirements of the platform.

6.2 Test Plan: A test plan that outlines the approach and strategy for testing the platform.

6.3 Test Scripts: Detailed scripts for each test case to be used for automated testing.

6.4 Test Data: Sample data to be used for testing different scenarios.

6.5 Test Reports: Reports summarizing the results of the tests and highlighting any discrepancies found.

6.6 Test Metrics: Metrics to measure the performance of the platform and the quality of the test results.

6.7 Test Documentation: Detailed documentation of the platform, including user guides, technical specifications, and release notes.

7. Test Acceptance Criteria

7.1 Suspension Criteria: Out of 100 percent of test cases if 40 percent of test case fail comes then it is suspension criteria until the development team fixes 40% of test cases fail.

7.2 Exit Criteria: If 95 percent of test cases out of 100 percent of test cases pass then that is the exit criteria.

8. Test Completeness

- 100% test coverage.
- All Manual & Automated Test cases are executed.
- All open bugs are fixed or will be fixed in the next release.