Functional Requirements Specification (FRS) Document

Application Name: Academy LMS Plugin

Document Version: 1.0

Date: 01-12-2024

Prepared By: MD SAKIL AHMED

Table of Contents

1. Introduction

This document outlines the functional requirements for the testing of the **Academy LMS Plugin** for WordPress. The plugin enables the creation and management of online courses and provides features like quizzes, course tracking, and user management. The objective is to ensure the plugin meets the requirements specified for its functionality, performance, and usability.

2. Scope of Testing

The scope of testing for the **Academy LMS Plugin** will include the following:

- **Functional Testing**: Testing of all core features and functionalities such as course creation, student registration, quizzes, and reporting.
- **Usability Testing**: Ensuring a seamless user experience for both administrators and students.
- **Compatibility Testing**: Testing plugin compatibility across different versions of WordPress and browsers.
- **Security Testing**: Verifying secure handling of sensitive user data (e.g., personal information, payment details).
- **Performance Testing**: Evaluating the plugin's performance under load (e.g., simultaneous users, large courses).

3. Functional Requirements

The following are the functional requirements for the **Academy LMS Plugin** to be tested:

3.1 User Registration and Management

- **FR1**: The plugin must allow users (students) to register on the platform via a user-friendly registration form.
- **FR2**: Administrators must be able to manage user accounts, including viewing, editing, and deleting student profiles.
- **FR3**: The plugin must support both manual and automated email notifications for user registration, login, and course enrollment.

3.2 Course Management

- **FR4**: Administrators must be able to create and manage courses, including adding course descriptions, lessons, and multimedia content.
- **FR5**: The plugin must allow administrators to assign instructors to courses and manage course schedules.
- **FR6**: The plugin must support setting different pricing models for courses (free, one-time payment, subscription).

3.3 Quizzes and Assessments

- **FR7**: Instructors should be able to create quizzes with various question types (multiple choice, true/false, short answer).
- **FR8**: The plugin must grade quizzes automatically and display results to students after completion.
- **FR9**: Administrators should be able to view detailed student quiz performance reports.

3.4 Course Enrollment and Tracking

- **FR10**: Students should be able to browse available courses and enroll in them easily.
- **FR11**: The plugin should track students' progress and display their current status (in progress, completed).
- **FR12**: Students should be able to resume courses they were previously enrolled in.

3.5 Payment Integration

- **FR13**: The plugin should integrate with popular payment gateways (PayPal, Stripe) for course purchase.
- **FR14**: Payment should be securely processed, and a confirmation email should be sent upon successful transaction.
- **FR15**: The system should handle failed payment attempts gracefully and notify users accordingly.

3.6 Reports and Analytics

- **FR16**: Administrators must have access to reports showing course completion rates, student enrollment statistics, and quiz scores.
- **FR17**: The plugin must allow the export of data to CSV or Excel format for further analysis.

4. Non-Functional Requirements

- **NFR1**: The plugin must load within 3 seconds in normal use cases.
- **NFR2**: The plugin must be compatible with the latest version of WordPress and at least the previous two versions.
- **NFR3**: The plugin must be mobile-responsive and work seamlessly on both desktop and mobile devices.
- **NFR4**: The plugin must be compatible with popular web browsers (Chrome, Firefox, Safari, Edge).
- **NFR5**: The plugin must handle concurrent users efficiently without significant degradation in performance.

---

5. Testing Scope and Types

5.1 Test Types

The following testing types will be performed:

1. **Functional Testing**: Verify that each feature listed in the functional requirements performs as expected.
2. **Integration Testing**: Ensure smooth interaction between the plugin and other WordPress plugins, themes, and external services (payment gateways).
3. **Usability Testing**: Ensure that the plugin provides a user-friendly interface for both administrators and students.
4. **Security Testing**: Verify data protection, secure login/logout processes, and encryption for sensitive information.
5. **Performance Testing**: Assess the system's performance under varying user loads using tools like JMeter.
6. **Compatibility Testing**: Test the plugin on different browsers, devices, and WordPress versions to ensure compatibility.

5.2 Test Environment

- **Operating System**: Windows/Linux/macOS
- **Web Server**: Apache/Nginx
- **PHP Version**: [Insert PHP version compatible with the plugin]
- **WordPress Version**: [Insert WordPress version]
- **Browsers**: Chrome, Firefox, Safari, Edge
- **Testing Tools**: Selenium WebDriver, JMeter, Postman, BrowserStack

6. Test Deliverables

- **Test Plan**: High-level strategy, objectives, and resources for testing.
- **Test Cases**: Detailed test cases for each feature.
- **Test Execution Reports**: Results of test case execution, including pass/fail status.
- **Bug Reports**: Documentation of defects found during testing with detailed reproduction steps.
- **Test Summary Report**: Summary of testing activities, including a final assessment of the plugin's readiness for release.

7. Assumptions and Dependencies

- The plugin is assumed to be installed and configured correctly on a WordPress website.
- The plugin's integration with external payment gateways will function as per their APIs and services.

8. Approval and Sign-off

This document will be approved by the project stakeholders (e.g., Product Manager, QA Manager, Development Team Lead) to proceed with testing.

This **FRS** serves as a guide to ensure comprehensive testing of the **Academy LMS Plugin**. It includes clear and testable functional requirements that can be used to create specific test cases and ensure all aspects of the plugin are tested. Let me know if you'd like to expand on any sections or if you need further customization!