Functional Requirements Document for Moodle Migration and Update

Table of Contents

1. Introduction

- 1.1 Purpose

- 1.2 Scope

- 1.3 Definitions, Acronyms, and Abbreviations

- 1.4 References

2. Overall Description

- 2.1 Product Perspective

- 2.2 Product Functions

- 2.3 User Classes and Characteristics

- 2.4 Operating Environment

- 2.5 Design and Implementation Constraints

- 2.6 Assumptions and Dependencies

3. Functional Requirements

- 3.1 Backup Current System

- 3.2 New Server Setup

- 3.3 Data Migration

- 3.4 System Update

- 3.5 Post-Update Verification

- 3.6 Maintenance and Support

4. Appendices

- 4.1 Appendix A: Backup and Restore Commands

- 4.2 Appendix B: Upgrade Commands

---

1. Introduction

1.1 Purpose

This document outlines the functional requirements for migrating and updating the Moodle Learning Management System (LMS). It aims to ensure a seamless transition to a new server and upgrade to the latest version of Moodle.

1.2 Scope

The scope of this document includes the assessment, backup, migration, update, and post-update verification processes for Moodle. It covers the necessary steps to ensure all data is transferred securely and the system operates correctly after the upgrade.

1.3 Definitions, Acronyms, and Abbreviations

- LMS: Learning Management System

- PHP: Hypertext Preprocessor

- SQL: Structured Query Language

1.4 References

- [Moodle Documentation](https://docs.moodle.org)

- [Moodle Download Page](https://download.moodle.org)

2. Overall Description

2.1 Product Perspective

The Moodle LMS is a widely-used platform for delivering educational content online. This project involves migrating Moodle to a new server and updating it to the latest version.

2.2 Product Functions

- Backup of the current Moodle system

- Migration of Moodle data and codebase to a new server

- Update of the Moodle system to the latest version

- Verification of system functionality post-update

- Ongoing maintenance and support

2.3 User Classes and Characteristics

- Administrators: Manage the Moodle system and perform updates.

- Instructors: Create and manage courses and content.

- Students: Access and participate in courses.

2.4 Operating Environment

- Server OS: Linux (recommended)

- Web Server: Apache or Nginx

- Database: MySQL, MariaDB, or PostgreSQL

- PHP: Version compatible with the latest Moodle release

2.5 Design and Implementation Constraints

- Ensure compatibility of plugins and themes with the new Moodle version.

- Minimize downtime during migration and update.

- Preserve existing data integrity and user settings.

2.6 Assumptions and Dependencies

- Access to both current and new servers.

- Availability of necessary backups.

- Adequate server specifications for the latest Moodle version.

3. Functional Requirements

3.1 Backup Current System

- FR1.1: Backup the Moodle database.

- FR1.2: Backup the Moodle data directory (moodledata).

- FR1.3: Backup the Moodle codebase.

3.2 New Server Setup

- FR2.1: Install required server software (PHP, database, web server).

- FR2.2: Create a new database for Moodle.

- FR2.3: Download and unpack the latest Moodle version.

3.3 Data Migration

- FR3.1: Copy the moodledata directory to the new server.

- FR3.2: Import the database backup into the new database.

- FR3.3: Transfer the Moodle codebase to the new server.

3.4 System Update

- FR4.1: Replace old Moodle codebase with the new version.

- FR4.2: Update custom plugins and themes.

- FR4.3: Run the Moodle upgrade script.

3.5 Post-Update Verification

- FR5.1: Verify system functionality.

- FR5.2: Test major functionalities (course creation, enrolment, plugins).

- FR5.3: Update configuration files.

3.6 Maintenance and Support

- FR6.1: Monitor system logs for errors.

- FR6.2: Ensure regular backups.

- FR6.3: Schedule regular Moodle updates.