Student Registration System SRS

1. Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for the Student Registration System. This system is intended to streamline the management of student registrations, course enrollments, teacher assignments, and related functionalities. The system will be used by administrators, students, and teachers, providing a comprehensive solution for managing academic processes.

1.2 Scope

The Student Registration System is a web-based application built using Laravel. The system includes modules for user authentication, course management, student enrollment, teacher assignments, and payment processing. The primary users of the system are administrators, students, and teachers.

2. Overall Description

2.1 Product Perspective

This system is a standalone application designed to manage student registrations and academic activities efficiently. It replaces manual processes with a web-based platform that integrates various functionalities.

2.2 Product Features

1. ***User Authentication***:
   - Registration and login functionality for students, teachers, and admins.
   - Role-based access control.

2. ***Course Management***:
   - Admin can create, edit, and delete courses.
   - Assign teachers to courses.

3. S***tudent Enrollment***:
   - Students can view available courses and enroll.
   - Admin can manage student enrollments.

4. ***Teacher Management***:
   - Admin can create, edit, and delete teacher profiles.
   - Assign teachers to specific courses.

5. ***Payment Processing***:
   - Students can pay registration fees and course fees.
   - Generate and display payment receipts.

6. ***Dashboard***:
   - Role-specific dashboards for admins, students, and teachers.

2.3 User Classes and Characteristics

- Admin:
  - Full access to all modules.
  - Manages courses, teachers, and student registrations.
  - Monitors payments and revenue.

- Student:
  - Registers and logs in to the system.
  - Views available courses and enrolls.
  - Makes payments and views receipts.

- Teacher:
  - Logs in to the system.
  - Manages assigned courses and tracks student assessments.

2.4 Operating Environment

- Database: SQLite
- Front-end: Laravel blade templates and Tailwind CSS for styling.
- Framework: Laravel
- Supported Browsers: Chrome, Firefox, Safari, and Edge.

2.5 Constraints

- The system relies on a functioning internet connection.
- Payments must integrate with a specified payment gateway.
- Limited to features explicitly provided in the project.

3. Functional Requirements

3.1 User Authentication

- Users can register and log in.
- Role-based access control (admin, student, teacher).

3.2 Course Management

- Admin can create, update, and delete courses.
- Admin can assign teachers to courses.

3.3 Student Enrollment

- Students can view and enroll in courses.
- Admin can manage student enrollments.

3.4 Teacher Management

- Admin can create, update, and delete teacher profiles.
- Admin can assign courses to teachers.

3.5 Payment Processing

- Students can pay registration fees and course fees.
- Payment receipts are generated and stored.

3.6 Dashboard

- Role-specific dashboards displaying relevant data.
- Admin: Overview of system metrics, payments, and user activities.
- Student: Enrolled courses and payment history.
- Teacher: Assigned courses and student assessments.

3.7 Assessment Management

- Teachers can create, view, and grade assessments.
- Scores are stored and accessible.

4. Non-functional Requirements

4.1 Performance Requirements

- The system must handle up to 1000 concurrent users.
- Response time for user actions should not exceed 2 seconds under normal load.

4.2 Security Requirements

- Passwords must be stored securely using hashing.
- Role-based access control to ensure data privacy.
- Use HTTPS for secure communication.

4.3 Usability Requirements

- The user interface must be intuitive and accessible.
- Provide clear error messages and guidance.

4.4 Maintainability

- The codebase must follow Laravel's coding standards.
- Documentation for setup and deployment should be provided.

4.5 Scalability

- The system should be scalable to accommodate growth in user numbers and data.