

# Software Requirements Specification (SRS)

## AI-Enabled Student Management System

### 1. Introduction

#### 1.1 Purpose

This document defines the requirements for the AI-Enabled Student Management System. The system aims to centralize and simplify access to student information while integrating AI to provide performance predictions and academic suggestions.

#### 1.2 Scope

- Students can view personal details, marks, syllabus, fee status, and faculty assignments
- Faculty and administrators can manage academic records
- AI-driven insights support academic performance

#### 1.3 Overview

- Web-based architecture
- Secure access and data management
- Integration of AI for academic assistance
- Streamlined experience for students and faculty

### 2. General Description

#### 2.1 Functions

- Student data management
- Marks and syllabus tracking
- Fee status updates
- AI-based performance prediction
- Faculty information and assignment management

#### 2.2 User Community

- Students
- Faculty
- Administrators

### **3. Functional Requirements**

#### **3.1 Possible Outcomes**

- Students view profiles, marks, syllabus, fees, and assigned faculty
- AI predicts academic performance and gives improvement suggestions
- Admins add or update academic data

#### **3.2 Ranked Order of Features**

- 1. Student Information Access
- 2. AI-Based Analytics
- 3. Admin Control Panel
- 4. Faculty Assignment and Updates

#### **3.3 Input-Output Relationships**

- Input: Student ID → Output: Student Profile
- Input: Marks Data → Output: AI Prediction & Suggestions

### **4. User Interface Requirements**

#### **4.1 Software Interfaces**

- Frontend: HTML, CSS, JavaScript
- Backend: Python (Flask/Django) APIs
- AI Integration: Machine Learning Models

#### **4.2 UI Examples**

- Student Dashboard: Cards for marks, syllabus, fees, and suggestions
- Admin Panel: Upload/edit academic records
- Notifications: Pop-ups for AI-generated suggestions

### **5. Performance Requirements**

#### **5.1 Response Time**

- Data fetch operations:  $\leq 2$  seconds
- AI results:  $\leq 5$  seconds

#### **5.2 Throughput**

Must support at least 50 concurrent users with no major degradation.

#### **5.3 Scalability**

Modular design to allow easy addition of users and new features.

## **6. Non-Functional Attributes**

### **6.1 Usability**

Simple, clean, and intuitive interface with minimal training required.

### **6.2 Reliability**

99% uptime with robust error handling and fault recovery.

### **6.3 Security**

- Role-based access control
- Data encryption
- Secure login/authentication

## **7. Schedule and Budget**

### **7.1 Timeline**

- Requirement Gathering: 1 Week
- Development: 4 Weeks
- Testing: 1 Week
- Deployment & Review: 1 Week

### **7.2 Cost Estimate**

Minimal cost due to use of open-source technologies and student effort.

## **8. Appendices**

### **8.1 Supplementary Information**

- AI algorithms considered:
- Linear Regression
- Random Forest
- Trained on historical student performance data

### **8.2 Glossary**

- SRS: Software Requirements Specification
- AI: Artificial Intelligence
- CRUD: Create, Read, Update, Delete
- UI: User Interface