# Software Requirements Specification (SRS)

## AI-Enabled Student Management System

### 1. Introduction

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 suggestions.  
  
Scope:  
The system allows students to view their details, marks, syllabus, fee status, and faculty assignments.   
Admins and faculty can manage academic records and view AI-based insights for academic support.  
  
Overview:  
The system will be web-based, secure, and AI-integrated. It will streamline academic management   
and assist both students and faculty through intelligent data-driven recommendations.

### 2. General Description

Functions:  
- Student data management  
- Marks and syllabus tracking  
- Fee status updates  
- AI-based performance prediction  
- Faculty management  
  
User Community:  
- Students  
- Faculty  
- Administrators

### 3. Functional Requirements

Possible Outcomes:  
- Student views profile, marks, syllabus, fees, and faculty info  
- AI predicts performance and gives suggestions  
- Admin adds/updates academic data

Ranked Order:

1. Student Information Access  
2. AI-based Analytics  
3. Admin Control Panel  
4. Faculty Assignment and Updates  
  
Input-Output Relationship:  
- Input: Student ID -> Output: Student Profile  
- Input: Marks Data -> Output: AI Prediction

### 4. User Interface Requirements

Software Interfaces:  
- Web-based front end (HTML, CSS, JS)  
- Backend APIs (Python/Flask or Django)  
- AI Module integration (using ML models)  
  
Examples:  
- Student dashboard with cards for marks, syllabus, fees  
- Admin panel for uploading and editing data  
- Pop-up or notification for AI suggestions

### 5. Performance Requirements

Response Time:  
System should respond within 2 seconds for data fetch and 5 seconds for AI results.  
  
Throughput:  
Must support 50+ concurrent users without performance degradation.  
  
Scalability:  
The system should allow for the addition of more users and modules without major restructuring.

### 6. Non-Functional Attributes

Usability:  
Simple, clean, and intuitive interface for students and staff.  
Reliability:  
99% uptime with proper error handling.  
  
Security:  
- Role-based access  
- Data encryption  
- Secure login authentication

### 7. Schedule and Budget

Timeline:  
- Requirement Gathering: 1 week  
- Development: 4 weeks  
- Testing: 1 week  
- Deployment & Review: 1 week  
  
Cost Estimate:  
As a student-level academic project, estimated development cost is minimal, using open-source tools.

### 8. Appendices

Supplementary Information:  
AI model may use algorithms like Linear Regression or Random Forest trained on past student performance.  
  
Glossary:  
- SRS: Software Requirements Specification  
- AI: Artificial Intelligence  
- CRUD: Create, Read, Update, Delete  
- UI: User Interface