Software Requirements Specification (SRS)

* Project Title: Learnytics- School Student Performance Tracker
* Version: 1.0
* Date: June 2025

# 1. Introduction

## 1.1 Purpose

The purpose of this project is to design and implement a centralized and automated system for tracking university student performance and attendance. The solution should allow stakeholders (academic coordinators, faculty, and counselors) to identify low-performing students, visualize subject-wise performance and attendance, and automate alerts and reports.

## 1.2 Scope

- Academic dashboard for subject-wise and student-wise insights  
- Pivot tables for grade and attendance visualization  
- Conditional formatting for performance tiers  
- Automated report generation (weekly/monthly)  
- Alerts for low attendance or failing grades  
- Integration capability with LMS platforms.

**2. Functional Requirements**

|  |  |
| --- | --- |
| ID | Requirement Description |
| FR1 | User Authentication for Admin, Faculty, and Academic Staff |
| FR2 | Upload and manage student records (Grades, Attendance %) |
| FR3 | Dynamic dashboards for subject-wise and student-wise insights |
| FR4 | Auto-highlight students with <40% attendance or failing grade |
| FR5 | Exportable reports (Excel/PDF) for departments |
| FR6 | Automated alert system via email/SMS |
| FR7 | Role-based access (Admin, Faculty, Analyst) |
| FR8 | Filters by Subject, Semester, Department |

# 3. Non-Functional Requirements

|  |  |
| --- | --- |
| ID | Requirement Description |
| NFR1 | Dashboard should load within 2 seconds |
| NFR2 | Secure backend with login encryption and HTTPS |
| NFR3 | Scalable to support 5,000+ student records |
| NFR4 | Fully responsive for mobile/tablet/desktop |
| NFR5 | Auto-refresh every 15 minutes for real-time updates |
| NFR6 | Data visualizations must follow data storytelling principles |

# 4. Constraints

|  |  |
| --- | --- |
| Component | Constraint |
| Database | PostgreSQL or MySQL |
| Frontend Framework | React.js |
| Backend | Python Flask/Django |
| Dashboard Tools | Power BI, Excel (for BA use) |
| LMS Integration | via REST API (optional) |
| Hosting Platform | Cloud-based (AWS/GCP suggested) |

# 5. Data Requirements

- Data Fields: Student Name, Subject, Grade (A, B, C, D, Fail), Attendance %  
- Data Sources: Internal academic records, Attendance systems  
- Data Quality: No duplicate student records, Grade values standardized, Attendance between 0-100%  
- Retention Policy: Student performance data retained for 5 academic years  
- Integrity: Grade entries linked with subjects; Attendance validated for range 0-100%

# 6. User Interface Requirements

- Frontend: React.js with Material UI or TailwindCSS  
- Dashboards: Admin Panel, Faculty Panel  
- Export: Excel, PDF, CSV  
- Cards: Avg Attendance, Top Performer, At-Risk List  
- Charts: Bar, Pie, Scatter  
- Accessibility: Keyboard nav, Color-blind support

# 7. Appendices

Admin Dashboard Sample Report

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Student ID | Name | Subject | Grade | Attendance % | Status |
| 202501 | A. Kumar | Data Science | C | 62% | Medium Risk |
| 202502 | R. Singh | Python | B | 85% | Safe |
| 202503 | S. D’Souza | Statistics | D | 38% | At Risk |