

Smart Student Management System (SSMS)


Mustack Ahamed

Department of Computer Science
Tagore College of Arts and Science, Chennai, India



<https://doi.org/10.55041/ijstmt.v2i5.042>

Cite this Article: Ahamed, M. (2026). Smart Student Management System (SSMS). International Journal of Science, Strategic Management and Technology, 02(05). <https://doi.org/10.55041/ijstmt.v2i5.042>

License:  This article is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting use, distribution, and reproduction in any medium, provided the original author(s) and source are properly credited.

Abstract— The Smart Student Management System (SSMS) is a comprehensive web-based application developed to automate and streamline student data management in educational institutions. Built with Python and the Flask framework, it provides a centralized platform for managing student profiles, tracking attendance, recording examination marks, and generating automated performance reports. By utilizing SQLite for storage and a three-tier architecture, the system ensures data integrity and scalability. This paper outlines the design, implementation, and testing of a system that addresses the inefficiencies and errors inherent in manual academic record-keeping.

Index Terms—Student Management, Flask, Web Development, Automation, Educational Technology.

I. INTRODUCTION

The rapid advancement of information technology has transformed educational administration. Many institutions still rely on traditional methods like physical registers and manual ledgers, which are time-consuming and prone to errors, data loss, and unauthorized access. These limitations are magnified in larger institutions where tracking thousands of students across multiple departments is required.

The Smart Student Management System (SSMS) is designed to digitize student records and provide intuitive interfaces for core administrative functions. It enables real-time performance monitoring, allowing faculty to identify at-risk students through automated attendance and grade reports.

II. SYSTEM SPECIFICATION

A. Hardware Specification

The SSMS is designed to run on standard commodity hardware, making it accessible for institutions with varying budgets.

- **Processor:** Intel Core i3 or higher (Min 1.6 GHz).
- **RAM:** Minimum 4 GB DDR4 (8 GB recommended).
- **Storage:** 250 GB HDD or SSD.
- **Network:** Broadband connection (Min 10 Mbps).

B. Software Specification

The system utilizes open-source technologies to eliminate licensing costs.

- **Language:** Python 3.13.
- **Framework:** Flask 3.0.
- **Database:** SQLite 3 with Flask-SQLAlchemy 3.1 ORM.
- **Frontend:** Bootstrap 5.3 and Jinja2 templating.
- **Hosting:** PythonAnywhere.

III. SYSTEM ANALYSIS

A. Existing System Drawbacks

Current manual systems suffer from several vulnerabilities:

- Time-consuming data entry and record-keeping.
- High probability of transcription and calculation errors.
- No real-time monitoring or remote access.
- Physical records are vulnerable to damage or loss.

B. Proposed System Advantages

The proposed SSMS provides a comprehensive digital platform with the following benefits:

- **Centralized Storage:** Eliminates physical record-keeping burdens.
- **Automated Calculation:** Grades are computed instantly based on predefined scales, removing human error.
- **Real-time Tracking:** Instant attendance percentage computation.
- **Scalability:** Architecture designed to accommodate growing student populations.

IV. SYSTEM DESIGN

A. Architectural Design

The system follows a three-tier architecture:

- **Presentation Tier:** Responsive web interface built with HTML5, CSS3, and Bootstrap 5.
- **Business Logic Tier:** Core logic in Python/Flask handling routing, validation, and calculations.
- **Data Tier:** Relational storage using SQLite.

B. Database Design

Three primary tables manage the data:

- **Student:** Stores name, unique registration number, department, and year.
- **Attendance:** Records status (Present/Absent) linked to a student ID and date.
- **Marks:** Stores subject-wise marks and auto-computed grades.

V. SYSTEM IMPLEMENTATION

A. Module Implementation

The Student Management module handles GET (form display) and POST (data storage) requests for adding or deleting students. The attendance route iterates through students for a selected date and saves records in a single transaction. The grade calculation module uses a predefined scale (e.g., **marks ≥ 90 = 'O'**, **marks ≥ 80 = 'A+'**, etc.) to assign grades automatically.

B. Deployment

The application is deployed on PythonAnywhere, ensuring accessibility across different devices and screen sizes.

VI. SYSTEM TESTING

Testing followed a Black Box approach to verify functional requirements.

- **TC 01 (Add Student):** Validated successful entry with valid data.
- **TC 02 (Duplicate Entry):** Verified that unique constraints on register numbers prevent duplicate records.
- **TC 09 (Attendance Accuracy):** Confirmed that attendance percentages are calculated correctly (e.g., 27/30 days = 90%).
- **TC 11-13 (Grade Assignment):** Verified that marks correctly map to grades (O, A+, F).

VII. CONCLUSION AND FUTURE WORK

The SSMS successfully automates academic administration, providing a professional and responsive tool for faculty. It eliminates the risks associated with manual data entry and provides reliable, on-demand reporting. Future enhancements will include user authentication, email notification systems, native mobile applications, and integration with institutional ERP systems.

VIII. REFERENCES

- [1] M. Grinberg, Flask Web Development: Developing Web Applications with Python, 2nd ed. O'Reilly Media, 2018.
- [2] A. Ronacher, Flask Documentation (Version 3.0), Pallets Projects, 2023. <https://flask.palletsprojects.com>
- [3] I. Sommerville, Software Engineering, 10th ed. Pearson Education, 2016.
- [4] R. S. Pressman, Software Engineering: A Practitioner's Approach, 8th ed. McGraw-Hill, 2014.
- [5] R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, 7th ed. Pearson Education, 2016.