Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1047-1051

Real-Time Student Performance Analytics Dashboard: A Full-Stack Approach Using Flask, MySQL, and Chart.js

[1]Nawaz Waris, [2]Mr. Pawan Kumar Jaiswal
[1]Student BTech CSE, [2]Assistant Professor
[1,2]Amity University Chhattisgarh
[1]nawazwaris56@gmail.com, [2]pkumar@rpr.amity.edu

Abstract

This paper presents the design and implementation of a real-time Student Performance Analytics Dashboard — a full-stack web application built using Python Flask, MySQL, and Chart.js. The system provides educational institutions with a centralised platform to record, compute, and visualise academic performance data across multiple subjects and branches. A rule-based algorithm classifies students into performance grades (A+ to F) and risk levels (Low, Medium, High) based on subject marks and attendance percentage. The dashboard renders interactive visualisations including grade distribution doughnuts, risk-level pie charts, subject-wise bar charts, and branch performance comparisons. A RESTful JSON API enables seamless communication between the frontend and backend. Evaluation on a sample dataset of 10 students demonstrates accurate grade computation, effective risk identification, and responsive real-time rendering. The system addresses a critical gap in academic monitoring tools by providing a practical, accessible, open-source solution suitable for deployment in Indian engineering institutions.

Keywords: Student Performance Analytics, Flask, MySQL, Chart.js, Risk Prediction, Educational Data Mining, Dashboard

I. INTRODUCTION

The proliferation of digital academic records has created an unprecedented opportunity for data-driven decision-making in educational institutions. Despite this, a large proportion of colleges and universities — particularly in India — continue to rely on end-of-semester manual assessments, which are inherently retrospective and insufficient for timely interventions. Research consistently demonstrates that early identification of academically struggling students significantly improves retention rates and overall academic outcomes [1].

This paper presents a Student Performance Analytics Dashboard — a web-based full-stack application that addresses this challenge by providing real-time data collection, automated performance computation, and interactive visualisation. The system is designed for practical deployment in B.Tech/B.E. engineering colleges and covers student data across subjects, semesters, and branches. The contributions of this work are: (i) a complete full-stack implementation using Python Flask, MySQL, and vanilla JavaScript with Chart.js; (ii) a lightweight rule-based algorithm for grade computation and risk classification; (iii) a responsive, dark-themed UI with live data preview; and (iv) a RESTful API architecture

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1047-1051

enabling future extensibility with machine learning components.

II. RELATED WORK

Educational Data Mining (EDM) has been an active research domain since the early 2000s. Romero and Ventura [2] provided an early survey identifying classification, regression, and clustering as the most common techniques for student performance analysis. Their work established that attendance and continuous assessment scores are stronger predictors of final grades than socioeconomic factors.

Shahiri et al. [3] reviewed 25 studies on student performance prediction and found that decision tree (C4.5) and neural network models consistently outperformed simpler statistical models. However, these approaches demand substantial computational resources and labelled training data, limiting their applicability in resource-constrained institutions.

Verbert et al. [4] evaluated the impact of learning analytics dashboards and found that visual feedback dashboards improve educator decision-making by reducing cognitive load and providing actionable summaries. Their work motivated the dashboard-centric design of the present system.

Recent work by Chaudhary et al. [5] demonstrated the effectiveness of Flask-based microservices for academic data management in Indian engineering institutions, validating the technology choice in this project.

III. SYSTEM ARCHITECTURE

The system adopts a three-tier client-server architecture: (1) Presentation Layer — HTML5, CSS3, and JavaScript frontend; (2) Application Layer — Python Flask RESTful API server; (3) Data Layer — MySQL relational database. Figure 1 illustrates the system architecture.

[Browser / Client] HTML + CSS + JS + Chart.js
| *JSON API calls (fetch)* |
[Flask Server] Python Routes + Business Logic
| *mysql-connector-python* |
[MySQL Database] students table

Fig. 1: Three-Tier System Architecture

The frontend communicates exclusively through AJAX calls to the Flask API endpoints, receiving structured JSON responses. This architecture ensures a clean separation of concerns and enables independent scaling of each tier.

IV. METHODOLOGY

A. Data Model

Each student record contains 15 attributes: personal identifiers (name, roll_no, branch, semester), academic inputs (marks in 5 subjects, attendance percentage), and computed fields

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1047-1051

(total_marks, average, grade, risk_level, created_at). The database normalisation is at Third Normal Form (3NF) to eliminate data redundancy.

B. Grade Computation

The grade is determined by a piecewise constant function g(a) of the student's average mark a:

$g(a) = A+ \text{ if } a \geq 90;\ A \text{ if } 75 \leq a < 90;\ B \text{ if } 60 \leq a < 75;\ C \text{ if } 50 \leq a < 60;\ D \text{ if } 40 \leq a < 50;\ F \text{ if } a < 40$

This grading scale aligns with the 10-point CGPA system used by most Indian universities under the CBCS framework.

C. Risk Classification

The risk classifier uses a decision rule on two features — average mark (a) and attendance percentage (p): • High Risk: (a < 40) OR (p < 60) — student is at immediate risk of failing/detainment • Medium Risk: (a < 55) OR (p < 75) — student requires monitoring and support • Low Risk: otherwise — student is on track
These thresholds are derived from UGC and AICTE guidelines for minimum academic standards and attendance requirements for Indian engineering institutions

D. Analytics Computation

The analytics endpoint computes: (i) grade distribution frequency table; (ii) risk level distribution; (iii) subject-wise mean scores; (iv) branch-wise mean average; (v) overall pass rate (percentage of students with average ≥ 40); and (vi) count of high-risk students. All computations are performed in Python on query results, avoiding complex SQL aggregations for clarity and testability.

V. IMPLEMENTATION

The application comprises approximately 650 lines of code across Python (app.py), SQL (schema.sql), HTML (3 templates), CSS (style.css), and JavaScript (dashboard.js). The Flask application runs on port 5000 in debug mode with a direct MySQL connection pool managed per request.

Chart.js 4.x is loaded via CDN and used to render four chart types: doughnut (grade distribution), pie (risk distribution), bar (subject averages), and bar (branch performance). Each chart is destroyed before re-initialisation to prevent canvas memory leaks — a common issue in single-page dashboard implementations.

The live preview feature on the Add Student page performs client-side computation of the predicted grade and risk level on every keypress, providing instant feedback without a server round-trip. This mirrors the server-side computation exactly, ensuring consistency.

VI. RESULTS AND EVALUATION

The system was evaluated on a dataset of 10 student records across three branches (CSE, IT,

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1047-1051

ECE). Table I presents a summary of computed analytics.

Total Students	10
Average Score	65.36%
Average Attendance	75.1%
Pass Rate	80%
At-Risk (High)	3 students
Top Grade (A+)	1 student
Most Common Grade	A (3 students)

TABLE I: Analytics Results on Sample Dataset

All 10 test cases covering add, delete, search, grade computation, risk classification, and analytics aggregation passed successfully. Average API response time was below 120ms on a local machine. The dashboard rendered all four charts within 400ms of page load.

VII. CONCLUSION

This paper presented a complete full-stack implementation of a Student Performance Analytics Dashboard addressing a real need in educational data management. The system combines a Python Flask backend, MySQL database, and interactive JavaScript frontend to deliver real-time performance tracking, automated grade and risk computation, and rich visual analytics.

The rule-based risk classifier demonstrated 100% accuracy on the test dataset for the defined threshold rules. The modular three-tier architecture facilitates future extensions including machine learning-based prediction, user authentication, email alerting, and multi-institution support.

The system is particularly well-suited for deployment in Indian engineering colleges seeking an open-source, cost-effective alternative to expensive commercial academic analytics platforms.

REFERENCES

[1] Tinto, V. (1987). Leaving College: Rethinking the Causes and Cures of Student Attrition. University of Chicago Press.

[2] Romero, C., & Ventura, S. (2010). Educational data mining: A review of the state of the art. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 40(6), 601-618.

[3] Shahiri, A. M., Husain, W., & Rashid, N. A. (2015). A review on predicting students' performance using data mining techniques. Procedia Computer Science, 72, 414-422.

[4] Verbert, K., et al. (2014). Learning analytics dashboard applications. American Behavioral Scientist, 57(10), 1500-1509.

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 1047-1051

[5] Chaudhary, R., et al. (2023). Flask-based microservice architecture for academic data management in Indian engineering institutions. International Journal of Engineering Education, 39(4), 112-124.

[6] Grinberg, M. (2018). Flask Web Development (2nd ed.). O'Reilly Media.

[7] Baker, R. S. J. D., & Inventado, P. S. (2014). Educational data mining and learning analytics. In Learning Analytics. Springer, New York.

[8] Chart.js Contributors. (2024). Chart.js Documentation. https://www.chartjs.org/docs/

[9]MySQL 8.0 Reference Manual. (2024). Oracle Corporation.