ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 05, May 2026

Early Prediction of Students' Academic Performance using Machine Learning

[1] Karishma R. Mujavar, [2] Samrudhi M. Bongale, [3] Siddhi S. Shah, [4] Anisha J. Shelake, [5] Dr. Deepali K. Jadhav

[1] [2] [3] [4] Computer Science & Engineering, KIT college, Kolhapur, Maharashtra, India
[5] Associate Professor, Computer Science & Engineering, KIT college, Kolhapur, Maharashtra, India
Corresponding Author Email: [1] karishmamujavar15@gmail.com, [2] bongalesamruddhi1218@gmail.com, [3] siddhishah2707@gmail.com, [4]anishashelake3646@gmail.com, [5] Jadhav.deepali@kitcoek.in

***Abstract*— In most colleges, a student's academic performance issues only come to light after the damage is already done. The general thing is to wait till end semester results, but this leaves very little time for improvement. The research done shows that there are 3 different types of data concerning how students' grades have changed, how their behavioural factors are, and how the socio-economic factors affect their learning. We use this to flag students who may be heading toward academic trouble, early enough to actually do something about it.**

The prediction model used is Logistic Regression, which produces a probability score for each student. This score is then classified into one of the four risk levels — Low, Medium, High, or Critical — using cutoff values of 0.25, 0.50, and 0.75. Four new features were created — a CGPI Progression Index tracking whether grades are going up or down over the years, a Study Focus Distribution checking how a student splits time across theory, practicals, assignments and revision, a Digital Accessibility Index based on whether the student has a phone, computer and internet at home, and a Socio-Economic Vulnerability Score built from family income and parental education levels. When a risk level is assigned, a built-in recommendation engine suggests specific actions the student or faculty should take. An AI layer using Groq's Llama 3.1 70B model is used to generate written explanations and study advice tailored to each student.

The whole system runs as a web application with separate views for students, teachers, and admins, built using FastAPI, MongoDB, and React. On training data, the model hit 93.54% accuracy, 99.58% ROC-AUC, and 98.32% recall. On unseen test data, accuracy held at 93.12% with ROC-AUC of 99.38%, showing the model generalises well without overfitting.

The reason for choosing Logistic Regression here is that it does not just perform well, it also explains itself. Faculty and admins can look at which factors drove a particular student's risk score, which builds trust in the system.

***Index Terms*— Academic Performance Prediction, Machine Learning, Early Intervention, Student Analytics, Risk Classification, Logistic Regression, Socio-Economic Risk Analysis.**

I. INTRODUCTION

The student's performance not only reveals the student's hard work and personality but also reflects the institution's teaching. There are many factors that actually shape the academic upbringing, which are not just limited to intelligence or efforts taken; a few other things included are the family income, internet availability, laptop or computer access, mental health, quiet environment for study, etc. Currently, institutions focus on waiting till final exams to identify the students who are struggling, which is too late to meaningfully intervene.

Everyone assumes that students who have been performing great since earlier will do the same ahead, but this is completely wrong; one must understand that the previous year's CGPA is not the only criterion to be concerned about. Those students from disadvantaged socio-economic backgrounds often show a decline in performance that their grade history gives no warning of. Hence, this makes it clear that relying on academic records is not enough; a broader picture of a student's situation is needed.

Socio-economic background is one of the most influential factors on how a student performs. Limited household income, parents with lower levels of education, lack of access to computers or the internet, and unstable home environments each independently affect a student's ability to engage in the lectures. So the prediction model that ignores these factors becomes unfair. It will miss the students who need support the most.

Machine learning combined with Educational Data Mining gives institutions a real opportunity to build systems with early-warning. These systems can bring together academic records, behavioural habits, and socio-economic data to produce risk estimates that are specific, timely, and actionable — rather than generic warnings that come too late.

This work makes the following contributions:

1. feature engineering technique that combines CGPA trends, study time distribution, internet access levels, and socio-economic status into a single unified risk score.
2. risk classifier that puts each student into one of four levels — Low, Medium, High, or Critical — using Logistic Regression, hitting 93.54% accuracy and 99.58% ROC-AUC, and simple understanding for non-technical staff as well.
3. recommendation engine that tells you exactly what to do once a risk level is assigned, along with an AI layer powered by Groq/Llama 3.1 that writes out

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 05, May 2026

personalized guidance for each student in simple language.

4. fully working web application with separate dashboards for students, faculty, and admins, which is built on FastAPI, MongoDB, and React, and ready to be deployed in an institution.

II. LITERATURE REVIEW

Machine learning is widely used as a tool for predicting student academic performances; it has various classification models been used in order to check performance across various educational settings. Logistic Regression is the most powerful starting point as it tells why we made a decision, and this matters when faculty or admin need to make a decision as an output for a student [1].

Decision Trees are another commonly used option since they produce rules that are easy to follow. The fallback here is that they memorize training data instead of learning from it, making them unreliable on larger, more complex student datasets. A better option than Random Forest is, one that addresses this by combining multiple trees, which generally leads to better results on unseen data. XGBoost pushes accuracy further by handling complex relationships between features, but at the cost of becoming harder to explain [4].

Other models like Multilayer Perceptrons and TabTransformer [3] have shown they can pick up patterns in structured student data that simpler models miss. But again, they need significantly more data to train properly, take longer to run, and they do not explicitly specify how they reach a conclusion; this makes institutions difficult to rely on for students' information.

When it comes to socio-economic factors, i.e., parental education, family income, and study environment play strong indicators of how students will perform. Studies suggest that there are a few institutes that support students to overcome the disadvantages to a lesser extent, but again, it's important to look at all the factors affecting them and not just grades. [6].

Recently, researchers have started collecting data from LMS (Learning Management System) that includes when students log in, when they submit the assignment, and the time spent on course videos, in order to predict performance [2]. But as many colleges do not have this data readily available, the best way is to use the self-reported information collected directly from students.

In all these existing works, there are a few gaps. Studies show that they focus on getting accurate numbers right, but say little about the students with low performance. There are very few systems that focus on socio-economic factors that could be used for real deployment. None of them uses an LLM layer to generate personaliz00ed guidance. And there is an absence of full working systems with separate interfaces for students, faculty, and admins. This project was built specifically to fill those gaps.

III. DATASETS

A. Primary Training Dataset

The training data was taken from the Mendeley Singapore Student Academic Performance Dataset, containing 12,411 student records across multiple academic batches. The dataset covers the following:

- **Academic Indicators:** Subject-level scores, professional domain competencies (PRO metrics), and a composite global score (G_SC)
- **Socio-Economic Features:** Father's and mother's education (EDU_FATHER, EDU_MOTHER), parental occupations (OCC_FATHER, OCC_MOTHER), household size (PEOPLE_HOUSE), and annual income (REVENUE)
- **Digital Access Indicators:** INTERNET, TV, COMPUTER, MOBILE
- **Demographic Attributes:** GENDER, QUARTILE The target variable is defined as: at_risk = 1 if QUARTILE == 1 (lowest performance quartile), else at_risk = 0

B. Real-Time Validation Dataset

Student data was gathered directly from enrolled students through a Google Form survey. Dataset link: https://docs.google.com/spreadsheets/d/1z0VV8kBAK1crXQaTGLZiWtftqDMI_cF06sNAuLwSmnQ/

The form was divided into four sections:

- **Demographic:** Name, Year of Study, Branch, College, Gender, Distance from home, Club participation, Career goal, Roll number
- **Academic:** 1st Year CGPA, 2nd Year CGPA, 3rd Year CGPA, Final Year CGPA
- **Socio-Economic:** Father's and mother's education and occupation, number of people in the household
- **Behavioural:** Job status, social media usage, stress level, assignment submission, attendance, study hours per day.

IV. METHODOLOGY

A. Data Preprocessing

Before the data could be used for training, it had to be cleaned up and brought into a consistent format. Here is what was done:

- Numbers that were missing were replaced with the **median** of that column, as median works better here because a few extreme values are handled, unlike mean.
- For text-based columns, blanks were filled with a fixed placeholder value; this way, "not filled" stays its own category instead of being confused with an actual answer.
- Text columns were then converted to numbers using **One-Hot Encoding**, with the first category dropped to

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 05, May 2026

avoid feeding the model redundant information.

- Columns like CGPA, income, and device usage were all on different scales, so **StandardScaler** was used to bring them to a common range.
- For outliers, the extreme values were simply pulled back to the nearest acceptable boundary using the **IQR method,** this way nothing was lost from the dataset.

Once all of this was done, the data was split into three parts: **70% for training, 15% for validation, and 15% for testing,** with stratified sampling making sure the proportion of at-risk students stayed consistent across all three.

B. Feature Engineering

Since raw features alone do not tell the full story, four additional features were constructed to capture aspects of a student's situation that would otherwise go unnoticed:

1. CGPI Progression Index: Rather than just looking at a student's latest CGPA, this feature looks at the direction their grades have been moving. A straight line is fitted across all four yearly CGPA values, and the slope of that line is used; if it is going down, that is an early warning sign worth acting on.

$$CGPI\_slope = \Sigma[(yi - \bar{y})(ti - \bar{t})] / \Sigma[(ti - \bar{t})^2]$$

2. Study Focus Distribution Vector: This one captures how a student spreads their study time across four areas: core subjects, lab and practical work, assignments, and revision. Each value is a percentage and all four add up to 100. The Shannon entropy of these four values is also worked out as a side feature — a very uneven split can be just as much of a concern as low study hours overall.

3. Digital Accessibility Index (DAI): A straightforward average of three yes/no indicators; whether the student has internet, a computer, and a mobile device at home. Students scoring low here are at a disadvantage when it comes to using online resources for studying.

DAI = (INTERNET + COMPUTER + MOBILE) / 3

4. Socio-Economic Vulnerability Score (SEVS): This score is built by combining normalised inverse values of father's education, mother's education, parental occupation, number of people in the household, and family income. Inverting these values means a higher final score points to greater disadvantage — making it easier to flag students who may need extra support beyond just academic help.

SEVS = mean([norm(EDU_FATHER_inv), norm(EDU_MOTHER_inv), norm(OCC_inv), norm(PEOPLE_HOUSE), norm(REVENUE_inv)])

C. Academic and Behavioural Scoring

When a student fills in their profile, the system calculates their risk score directly from that data instead of only depending on the ML model. Each input gets converted into a score between 0 and 100, and these scores are then combined to get the final risk value.

Academic Score: The CGPA is first checked to figure out which scale it is on. If it is out of 10, it gets multiplied by 10. If it is out of 4, it gets multiplied by 25.

Attendance Score: If attendance is given as a range like 50–75%, the midpoint is taken. If it is given as a lower bound like >90%, 5 is added to that value and capped at 100.

Study Hours Score:

- 6+ hrs/day → 85
- 3–6 hrs/day → 65
- 1–3 hrs/day → 40
- If an exact number is given → min(100, n×15)

Stress Score: Stress is collected on a scale of 0 to 5. It is then converted using: Stress_Score = 100 − (stress / 5) × 80

So All Together:

- Behaviour_Score = 0.40 × Attendance + 0.35 × Study Hours Score + 0.25 × Stress Score
- Composite = 0.50 × Academic Score + 0.50 × Behaviour Score
- Risk_Score = (100 − Composite) / 100

D. Risk Classification

The final risk score is mapped to one of four levels:

Risk Level	Condition	Action
Low	Score < 0.25	Performance Optimization
Medium	0.25 ≤ Score < 0.50	Structured Support
High	0.50 ≤ Score < 0.75	Active Remediation
Critical	Score ≥ 0.75	Priority Intervention

When the system runs predictions for an entire class at once, a ranking-based adjustment is applied to make sure the results are realistic. Without this, in a strong performing batch, every student might end up with a Low Risk label, which does not really help anyone. So the system looks at the relative standing of students within that group and ensures a reasonable number get flagged

across the higher risk levels.

E. Model Development

Primary Model: Logistic Regression: Chosen for three practical reasons as it explains its decisions through coefficients, it runs fast enough for real-time use, and it directly outputs a probability score between 0 and 1. The model was trained with balanced class weights to handle the fact that at-risk students are fewer in number. Optimisation was done using L-BFGS with binary cross-entropy as the loss function.

Deep Learning Extension: MLP (PyTorch): An optional deeper model was also built with the following layer structure: Linear → ReLU → Dropout(0.3) → Linear(256) → ReLU → Dropout(0.3) → Linear(128) → ReLU → Linear(1) → Sigmoid

Trained using Adam optimiser with binary cross-entropy loss.

Transformer Extension: A TabTransformer-based structure was set up as a future extension to handle categorical features through contextual embeddings.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 05, May 2026

F. Evaluation Metrics

- Accuracy = (TP + TN) / (TP + TN + FP + FN)
- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)
- F1-Score = 2 × (Precision × Recall) / (Precision + Recall)
- ROC-AUC = Area under the ROC Curve

G. System Architecture

The system is divided into three layers:

Frontend (React + TypeScript + Vite):

- Three separate dashboards were built depending on who is logging in.
- Students get to see their own risk level, how their CGPA has changed over the years, how they are spending their study time, their behavioural habits, and AI-generated insights specific to their profile.
- Faculty members can see the full list of their students along with attendance and marks, and can directly assign remedial actions to whoever needs them.
- Admins get a broader view of how the entire batch is doing, can approve or reject student registrations, and manage which users have access to what within the system.

Backend (FastAPI): Consists of a set of API endpoints to handle functions like login and authentication, student profile management, risk prediction, remedial recommendations, LLM-based insights, faculty messaging, and admin controls. A healthy checkpoint tells that ML models work correctly.

Database (MongoDB): Has five collections that store user accounts, student profiles, messages, remedial assignments, and faculty profiles. Each student gets a unique ID generated from the first 8 characters of their email's SHA-256 hash, this helps to keep identities consistent without personal identifiers.

V. AI-POWERED RECOMMENDATION & INSIGHTS

A. Remedial Recommendation Engine

Once a student's risk level is assigned, the system suggests it a set of recommended actions:

For students marked as **Critical or High Risk**, the response is more intensive. They are put into mandatory tutoring sessions focused on the subjects they are struggling with, assigned a faculty mentor who checks in every two weeks, referred for counseling if stress is a factor, and given a structured emergency academic plan with clear checkpoints to track progress.

For **Medium Risk** students, the approach is a bit lighter. They are directed toward time management workshops, encouraged to join peer study groups, given help with weekly study planning, and offered optional counseling if they feel they need it.

For students in the **Low Risk** category, the focus shifts from damage control to growth. They are given guidance on how to push their performance further, pointed toward advanced learning material and research opportunities, and supported in preparing for competitive exams if that is a goal they have.

B. LLM-Augmented Personalized Insights

Along with the rule-based engine, the Student Dashboard includes an AI Insights section that generates responses specific to each student's profile rather than showing the same advice to everyone.

1. **Explain My Risk:** When a student wants to understand their risk score, they can request an explanation through the Explain My Risk feature. The system picks up their academic scores, behavioural data, and risk tier, sends it to Groq's Llama 3.1 70B, and returns a written explanation of what is driving their score.
2. **Personalized Study Tips:** It looks at the particular student's CGPA trend, attendance, study hours, and stress

levels, and home background, and puts together advice that actually fits their situation — whether that is subject-specific revision strategies, stress management, or making better use of available digital resources.

VI. RESULTS AND DISCUSSIONS

Training Data Performance:

Performance Metric	Value
Accuracy	93.54%
Precision	56.22%
Recall	98.32%
F1 Score	71.53%
ROC-AUC	99.58%

Testing Data Performance:

Performance Metric	Value
Accuracy	93.12%
Precision	54.71%
Recall	98.05%
F1 Score	70.23%
ROC-AUC	99.38%

The gap between training and test accuracy was just 0.42%, which means the model is picking up actual patterns rather than just fitting itself to the training data. The ROC-AUC crossing 99% on both sets is a good sign too — it means the model holds up well regardless of where the decision threshold is set.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 05, May 2026

Out of all the metrics, recall mattered the most. Missing a student who genuinely needs support is a much bigger problem than accidentally flagging someone who ends up being fine. Getting recall above 98% means the model is catching almost every at-risk student. The trade-off is lower precision, but that was a deliberate choice — it is better to have a few unnecessary check-

ins than to leave a struggling student unnoticed.

Adding socio-economic features turned out to make a real difference, especially for students from weaker financial backgrounds. Models that only look at grades often miss this group entirely, so having that extra layer helped the system be more fair and more complete in who it flagged.

VII. CONCLUSION AND FUTURE WORK

The project solves the problem that most institutions don't take seriously. Our goal was to find out students at risk early enough to actually help them, instead of waiting til their end-of-semester exams. The system uses a combination of academic history, behavioural patterns, and socio-economic background to flag students who may be at risk, while there is still time to step in.

The model used, i.e., Logistic Regression, hit 93.54% accuracy and 99.58% ROC-AUC; these numbers are good for this kind of task. Importantly, it not only predicts but explains as well. Faculty and admins can see which factors lead to students with poor academic performance, making the system more trustworthy rather than just producing labels.

Once a risk level is assigned, the recommendation engine steps in and suggests what needs to be done next. A critical risk student gets directed toward mandatory tutoring, while a low-risk student might just be pointed toward competitive exam prep. The Groq/Llama 3.1 layer makes this even more useful by writing out explanations and study tips that are built around that specific student's data, rather than being the same for everyone.

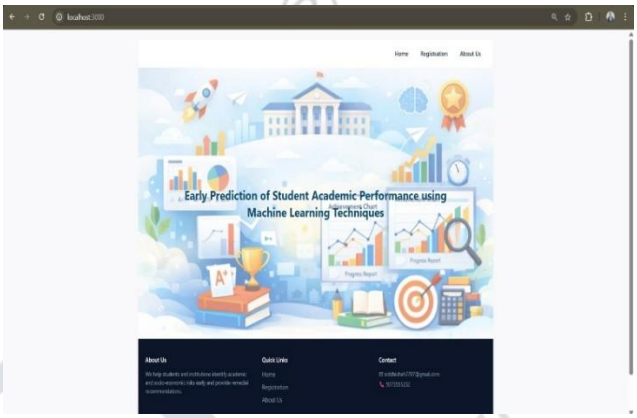
On the technical side, everything is in place for this to be deployed in a real college setup. The FastAPI backend handles all the logic, MongoDB stores the data, and the React frontend gives students, faculty, and admins their own separate spaces to work from. Together, they form one connected platform where prediction, approvals, analytics, and communication all happen in the same place.

Future development directions include:

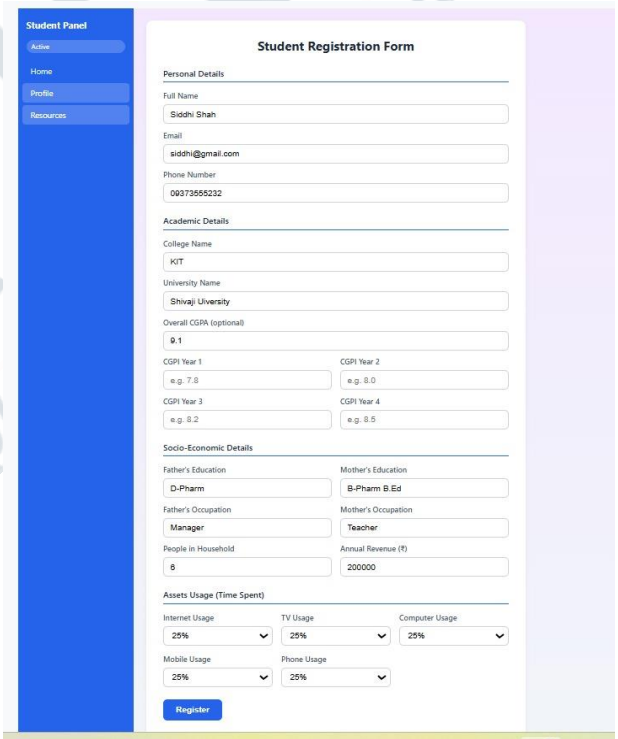
- Bringing in the TabTransformer architecture [3] to better handle relationships between categorical socio-economic variables
- Adding SHAP-based explainability for per-student feature attributions beyond global model coefficients
- Connecting to a Learning Management System for real behavioural data like assignment submissions and quiz scores [2]
- Scaling deployment using container orchestration to support larger institutional populations
- Extending the model from binary at-risk prediction to multi-class outcomes like dropout or course failure
- Tracking student response to interventions and feeding that data back into model retraining cycles.

VIII. OUTPUTS

A. Landing Page



B. Student Registration Page

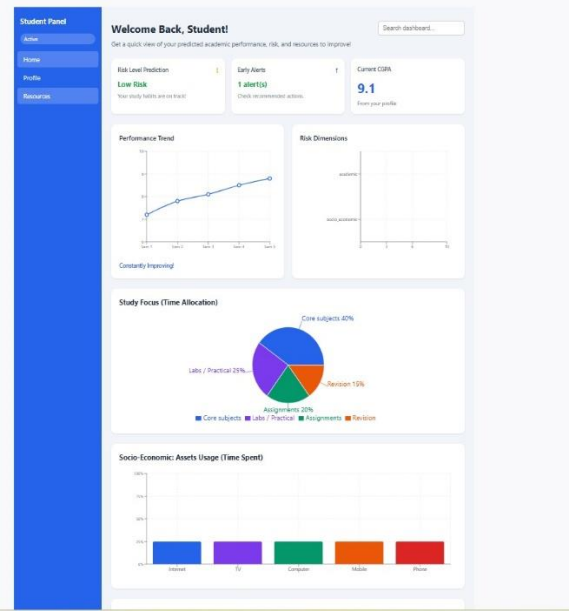


ISSN (Online) 2394-2320

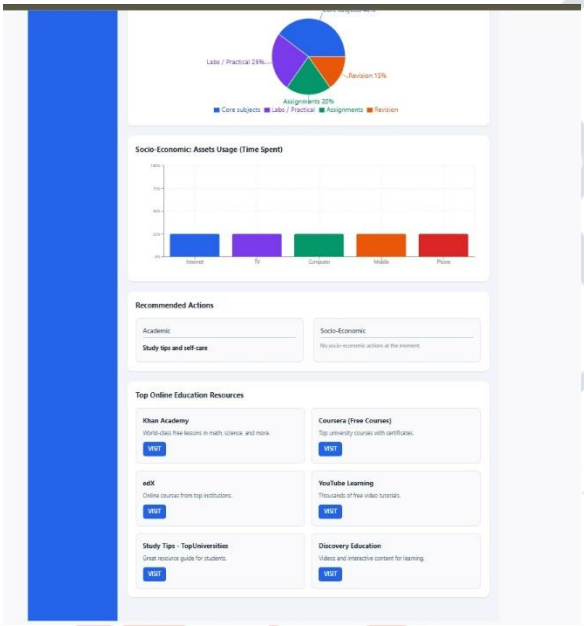
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 05, May 2026

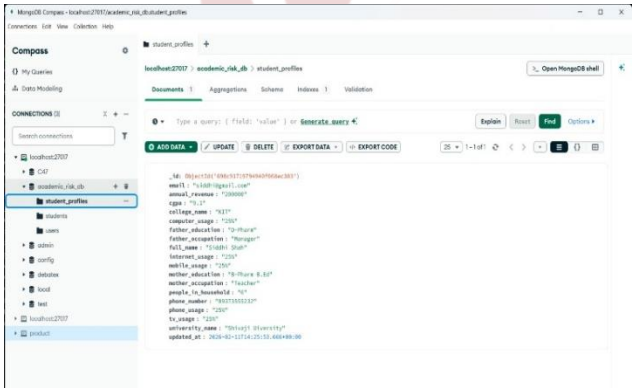
C. Student Dashboard (View 1)



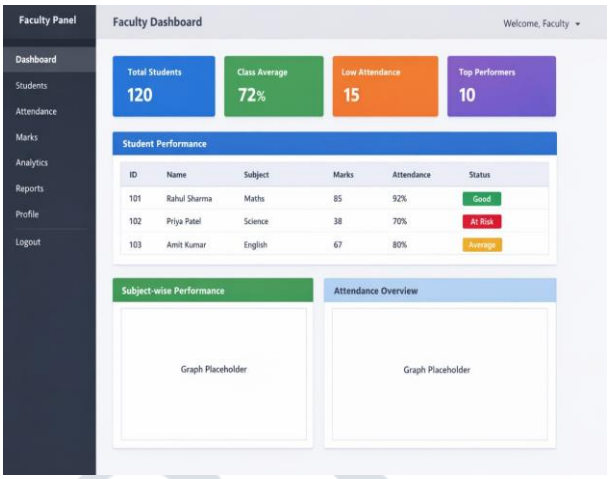
D. Student Dashboard (View 2)



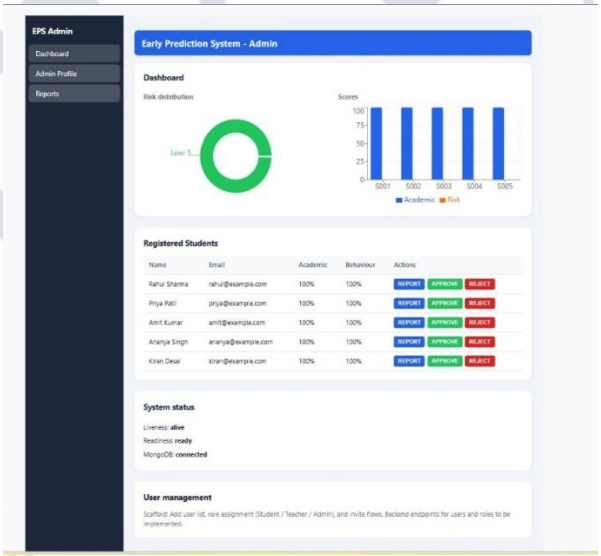
E. Database View



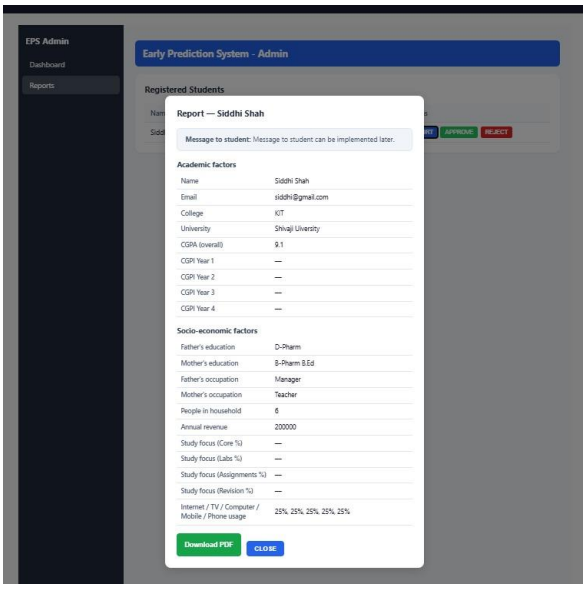
F. Faculty Dashboard



G. Admin Dashboard



H. Student Report



REFERENCES

[1] C. Dervenis, V. Kyriatzis, S. Stoufis, and P. Fitsilis, "Predicting students' performance using machine learning algorithms," in *Proc. 6th Int. Conf. on Algorithms, Computing and Systems (ICACS)*, ACM, 2022

[2] J. Wang and Y. Yu, "Machine learning approach to student performance prediction of online learning," *PLOS ONE*, vol. 20, no. 1, p. e0299018, 2025

[3] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin, "TabTransformer: Tabular data modeling using contextual embeddings," arXiv:2012.06678, 2020

[4] A M. Rahman et al., "XGBoost for educational analytics,"*Journal / Conference*, 2022

[5] Ali Shoukat, Zubair Haider, Hamid Khan and Awais Ahmed, "Factors Contributing to the Students Academic Performance: A Case Study of Islamia University Sub- Campus." American Journal of Educational Research 1, No.8 (2013):283-289 doi: 10.12691/education-1-8-3

[6] A. Verma and P. Sharma, "Student Performance Prediction using AI and ML," *International Journal of Computer Applications*, vol. 182, no. 45, pp. 12–18, 2023. (Student_Performance_Prediction_using _AI.pdf)

[7] A. K. Jaiswal and P. Patil, "Student General Performance Prediction using Machine Learning," *International Journal of Engineering Research & Technology*, vol. 12, no. 6, pp. 1–6, 2023. (STUDENT_GENERAL_PERFORMANCE_PREDICTION_U.pdf)

[8] S. Gupta and R. Tiwari, "Student Performance Prediction Using ML Model," *International Research Journal of Engineering and Technology*, vol. 10, no. 3, pp. 4561– 4567, 2023. (STUDENT PERFORMANCE PREDICTION USING ML MODEL.pdf)

[9] R. L. Kumar and S. N. Singh, "Student Performance Prediction Using Machine Learning Algorithms: A Review," *Student Performance Prediction Using Machine Learning Algorithms*,pp. 1–7, 2023. (Student_Performance_Prediction_using_Mac.pdf)

[10] M. Ali, F. Rahman, and K. Hussain, "Student Performance Prediction in Higher Educational Institutes Using Multi-Swarm PSO," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 2, pp. 50–58, 2023. (Student_Performance_Prediction_in_Higher.pdf)