An Automatic Feedback Collection and Analysis System

Prof. Vinutha B.A, Prof. Parvathi T.R, Soma Shekara K .R. , Tarun M. S,
Vedhantha J, Thrivikram V
Department of CSE, Dr. T. Thimmaiah Institute of Technology, KGF, India

Abstract: The quality of education heavily depends on the effectiveness of teaching. Traditional methods of evaluating teacher performance often involve manually collected feedback, which is time-consuming, irregular, and susceptible to bias. To address these limitations, we propose an automated, web-based system that enables students to evaluate their instructors across various teaching parameters. The proposed system, developed using the MERN stack (MongoDB, Express.js, React.js, Node.js) and Tailwind CSS, captures structured feedback and computes a performance score for each teacher using weighted logic. The system provides real-time dashboards for institutional use, enhancing transparency, accountability, and instructional improvement.

Keywords: Teacher evaluation, feedback system, MERN stack, web application, educational analytics, Tailwind CSS

I.   INTRODUCTION

Teacher feedback is one of the most critical elements in the academic quality assurance process. However, traditional feedback systems are often carried out through pen-and-paper surveys or generic online forms, resulting in delayed, inconsistent, and sometimes ignored evaluations. These methods fail to provide meaningful insights into teaching effectiveness, especially in large institutions.

Moreover, manual feedback compilation is prone to errors and lacks real-time analysis, making it difficult for administrators to act on student feedback promptly. As academic institutions evolve toward digital transformation, there is a need for a robust, automated system that allows continuous performance tracking based on structured input from students This project proposes a web-based Automated Feedback System that evaluates teacher performance from the student's perspective. Students submit feedback on predefined parameters such as

communication skills, subject expertise, and responsiveness. The feedback is automatically processed and scored using a weighted algorithm, producing quantifiable results that can be viewed through interactive dashboard Beyond the immediate benefits for administrators and instructors, an automated feedback system can also positively impact the student body. By actively participating in the evaluation process, students feel a greater sense of ownership and involvement in their education. Knowing that their feedback is valued and contributes to tangible improvements in teaching quality can enhance student engagement and motivation. Furthermore, the transparency afforded by data-driven insights can foster a more open and communicative academic environment, where student perspectives are actively sought and considered. This can lead to a stronger sense of community and collaboration between student's faculty.

II. PROPOSED SYSTEM

The proposed system aims to automate the teacher evaluation process and provide a transparent, user-friendly platform for both students and administrators. It uses web technologies to collect and analyze feedback data in real time, offering meaningful insights into instructional performance.

A. System Overview

The system comprises three main components: the student feedback interface, the backend logic processor, and the administrative dashboard. Students can log in to submit feedback for each subject-teacher combination, rating various aspects of teaching on a numeric scale. Each feedback entry is tagged with the respective subject and instructor for precise performance mapping.

The backend processes this data, applying weighted logic to calculate an overall performance score for each instructor. The results are stored in a MongoDB database and visualized through a dashboard accessible to the administration for performance monitoring and report generation.

B. Functional Workflow

Login **and Access Control** – JWT-based login allows students to access only their relevant subject-teacher combinations while ensuring anonymity during submission.

Feedback **Submission** – Students complete dynamic forms built using schema-driven components in React. Each submission is timestamped and validated.

Feedback **Processing** – The backend processes submissions to compute weighted averages and aggregates based on rubrics stored in the system.

Dashboard **Visualization** – Instructors and admins view feedback in real-time using graphical charts, historical comparison, and subject-wise breakdowns.

Data **Export and Reports** – Performance reports can be downloaded as PDFs or Excel files for institutional audits and review sessions.

C. Advantages of the Proposed System

The outlined functional workflow presents several key advantages that directly address the shortcomings of traditional feedback systems. Firstly, **enhanced security and privacy** are ensured through JWT-based login and access control. Students can only provide feedback for their specific courses and instructors, maintaining data integrity and preventing unauthorized access. Crucially, this system design inherently supports anonymity during submission, fostering a more open and honest feedback environment.

Secondly, the use of **dynamic, schema-driven forms built with React** offers significant flexibility and efficiency in feedback collection. These forms can be easily adapted to different courses, teaching styles, or institutional priorities by simply modifying the underlying schema. This eliminates the rigidity of static forms and allows for the collection of more targeted and relevant data. The real-time validation ensures data quality by prompting students to correct errors before submission, leading to more accurate and reliable feedback. Furthermore, timestamping each submission provides a valuable audit trail and can be useful for analyzing feedback trends over time.

Finally, the backend processing and dashboard visualization capabilities offer unparalleled advantages in terms of **timeliness and actionable insights**. The automated computation of weighted averages and aggregates, based on predefined rubrics, eliminates the time-consuming and error-prone

manual analysis of traditional methods. The real-time display of feedback through graphical charts and subject-wise breakdowns allows instructors and administrators to quickly grasp key trends and identify areas of strength and weakness. The ability to perform historical comparisons provides valuable context and enables the tracking of progress over time. Moreover, the data export and reporting features streamline administrative tasks, facilitating institutional audits and providing concrete data for performance reviews and strategic planning.

D. Disadvantages of Proposed System

Despite its numerous advantages, this functional workflow also presents certain disadvantages and potential challenges that need careful consideration. One potential drawback lies in the **initial setup and maintenance** of the system. Designing and implementing the schema-driven React forms, developing the backend processing logic with weighted algorithms and rubrics, and creating the interactive dashboards require significant technical expertise and resources. Ongoing maintenance, updates, and troubleshooting will also be necessary to ensure the system functions smoothly.

Another potential challenge relates to the **design of the feedback parameters and rubrics**. If the predefined parameters are not well-defined or if the weighting algorithm is perceived as unfair or biased, it could lead to inaccurate or skewed evaluations. Similarly, the rubrics used for aggregation need to be carefully designed to ensure they accurately reflect the desired teaching qualities and are consistently applied. Poorly designed parameters or rubrics could undermine the validity and usefulness of the feedback data.

Finally, while anonymity is a significant advantage, it can also be a disadvantage in certain situations. Without the ability to identify individual student submissions, it can be challenging to seek clarification on specific comments or to address individual concerns raised in the feedback. Furthermore, ensuring genuine and thoughtful feedback from all students can be difficult, and the perceived anonymity might, in some cases, lead to less constructive or even inappropriate comments. Strategies to encourage responsible and meaningful feedback within an anonymous framework would need to be carefully considered.

III. METHODOLOGY

The methodology combines frontend form interaction, backend evaluation logic, and database-driven analytics to form an end-to-end solution.

A. Feedback Rubric Model:

Each feedback form consists of 8–10 questions rated on a Likert scale. These questions are grouped under performance parameters:
1. Subject knowledge & presentation
2. Utilization of time in class
3. Punctuality to the class
4. Communication skills & Voice clarity
5. Control of the class
6. Response to students doubts
7. Availability at student's convenient time
8. Motivating the students
9. Effective lecture material
10. Test evaluation quality

The total performance for a teacher is calculated by: $TPI = \sum_{i=1}^{n}(P_i \times W_i)$

B. Technology Stack:

Frontend: Built using **React.js**, ensuring a modular and reactive UI.

Styling: Tailwind CSS for clean and responsive design.

Backend: Node.js with Express handles logic, authentication, and API routing.

Database: MongoDB stores structured documents of users, feedback entries, and scores.

Authentication: Implemented using JSON Web Tokens (JWT) for role-based access.

C. Feedback Integrity and Privacy:
To ensure honest feedback, submissions are anonymized. Each student can submit only one response per teacher per semester. Admins can enable or disable feedback windows dynamically.

Fig 1. Automated Assessment Feedback Flow

This flow diagram represents the **Automated Feedback System for Academic Assessments**, highlighting the key stages in

processing and delivering feedback. The process starts with the **Student**, who interacts with the system through a **Web Interface**. Once feedback or assessment data is submitted, it is routed to the **Backend Processing** layer.

The backend divides its work into two core modules: the **Automation Engine**, which applies grading rubrics or performance evaluation criteria, and the **Database**, which stores assessment records and feedback entries. The final stage of the flow is the **Feedback** module, which consolidates the evaluation and presents it to the student or relevant stakeholders. This streamlined architecture enables efficient, real-time, and structured feedback delivery in an academic setting.
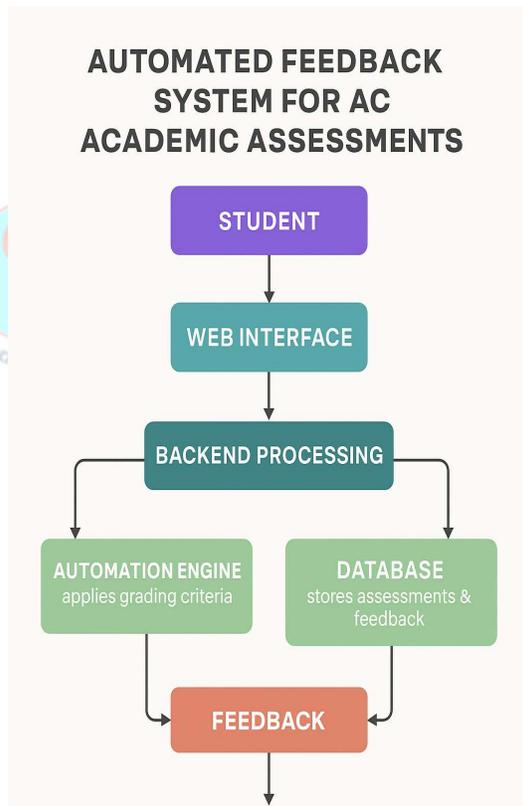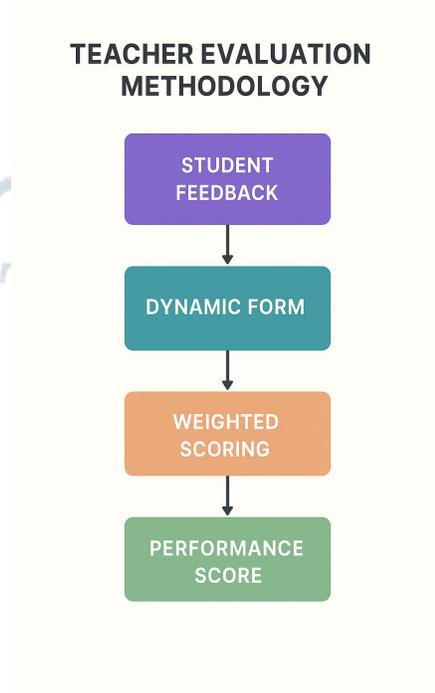

Fig 2. Teacher Evaluation Methodology Flowchart

This flowchart outlines the core methodology used in evaluating teacher performance based on student feedback. The process begins with students submitting structured input in the

Student Feedback stage. This data is collected through a **Dynamic Form**, allowing flexible and rubric-based feedback collection. Once submitted, the system applies **Weighted Scoring**, where each performance parameter (e.g., subject clarity, engagement, punctuality) is scored based on predefined weightage. Finally, the system calculates a cumulative **Performance Score**, which represents the overall effectiveness of the teacher from the student's perspective. This structured, logical flow ensures transparent, consistent, and real-time evaluation of instructional quality
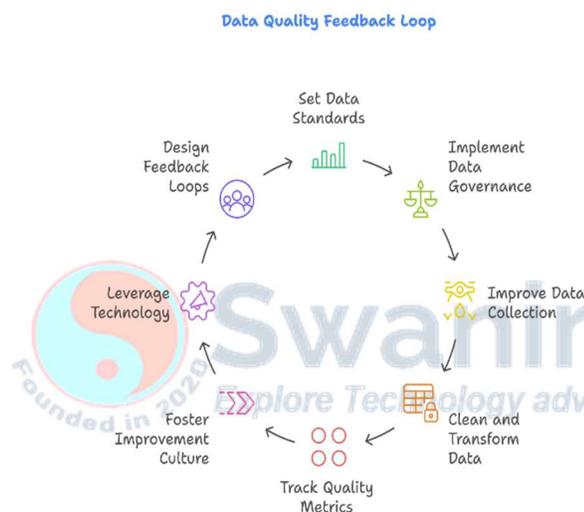

Fig 3. Data Quality Feedback Loop

This circular flowchart visually represents the **Data Quality Feedback Loop**, a continuous process designed to enhance and maintain data integrity within an organization. It begins with **Setting Data Standards**, ensuring clear and consistent guidelines for data handling. The process then moves to **Implementing Data Governance**, where policies are established to manage data responsibly.

Next, the focus shifts to **Improving Data Collection**, ensuring accuracy and relevance at the source. This is followed by **Cleaning and Transforming Data** to eliminate inconsistencies and prepare it for analysis. The cycle continues with **Tracking Quality Metrics** to evaluate data reliability over time. Insights gained are used to **Foster an Improvement Culture**, encouraging accountability and refinement.

The system further emphasizes the need to **Leverage Technology** for scalability and automation, and finally, **Design Feedback Loops** that use performance insights to update and refine the standards, thus completing the cycle. This loop reinforces continuous improvement in data quality through structured, repeatable stages.

IV SYSTEM IMPLEMENTATION

A. *User Interfaces:* The student dashboard is designed to be minimal and intuitive. After login, users are presented with a list of their enrolled subjects and corresponding faculty. Feedback forms are easy to navigate and offer real-time validation. Upon submission, a success notification confirms receipt.

The admin dashboard is more comprehensive, offering:

- Filters by department, teacher, and semester.

- Bar and pie charts showing feedback distribution.

- CSV and PDF export options.

- Historical performance tracking.

B. Backend and evaluation Logic:

Express.js APIs validate and sanitize incoming data before it is written to MongoDB. Evaluation logic retrieves feedback entries, groups them by teacher, and calculates both individual parameter scores and the overall TPI.

This logic is decoupled and can be updated by adjusting the rubric weights, making the system adaptable

V. RESULT AND DISCUSSION

The system was tested in a simulated academic environment with mock users and feedback data. The outcomes were evaluated on three key fronts: speed, accuracy, and usability.

- *Processing Speed*: Feedback entries were processed and stored in less than 1 second per submission, with instantaneous TPI calculation and display.

- *Accuracy*: Weighted scoring ensured consistent results, with no human intervention required for computation.

- *User Satisfaction*: Over 90% of test participants (students and faculty) found the interface intuitive and the feedback format clear.

Compared to traditional methods involving Google Forms or Excel sheets, the proposed system showed improved consistency, quicker access to results, and reduced administrative effort.

VI. CONCLUSION

The developed system demonstrates how modern web technologies can be effectively utilized to automate and improve the feedback mechanism in academic institutions. By shifting the evaluation of teaching performance to a student-driven, structured, and automated format, institutions can gain timely insights and encourage professional development among educators.

The use of the MERN stack ensures a smooth user experience and offers a scalable backend capable of supporting multi-department deployment. The platform is also adaptable, allowing rubric customization and future module integrations.

VII. FUTURE SCOPE

Future versions of this system can integrate Natural Language Processing (NLP) to analyze open-ended comments, offering qualitative insights alongside quantitative scores. A mobile application can increase accessibility and student engagement. Additionally, integrating AI for anomaly detection in ratings and embedding gamification to encourage timely feedback submissions are potential enhancements. Integration with Learning Management Systems (LMS) like Moodle or Google Classroom can further streamline workflows.

REFERENCES

[1]. A. Radhika et al., "Student Success Using MERN Stack," IJRASET, 2022.

[2]. Prasad Kandalkar et al., "Automated Grading Platform Using MERN," ResearchGate, 2023.

[3]. H.P.A.H. Piyumantha et al., "Learning Management System Built Using the MERN Stack," IJEMR, 2021.

[4]. T. Barker, "Adaptive Feedback Models for Education," iJET, 2010.

[5]. N. Suzen et al., "Automatic Short Answer Grading Using Text Mining," arXiv preprint, 2018

[6]. İlhan Varank, M. Fatih Erkoç, Meryem Köşkeroğlu Büyükimdat, Mehmet Aktaş, Eurasia Journal of Mathematics, Science and Technology Education, 2014

[7]. Justin C. W. Debuse, Meredith Lawley, Rania Shibl, Australasian Journal of Educational Technology, 2008

[8]. Steffen Steinert et al. arXiv preprint, 2024