# Unlocking subjective answer assessment using ML and NLP

Prof. Mrs. Vidya Nemade<sup>1</sup>, Shruti Patil<sup>2</sup>, Ishwari Patil<sup>3</sup>, Pallavi Ghatwal<sup>4</sup>, Aanchal Oswal<sup>5</sup>

<sup>1</sup>Prof, P.E.S.Modern college of engineering, Pune <sup>2</sup>Students, P.E.S.Modern college of engineering, Pune <sup>3</sup>Students, P.E.S.Modern college of engineering, Pune <sup>4</sup>Students, P.E.S.Modern college of engineering, Pune <sup>5</sup>Students, P.E.S.Modern college of engineering, Pune

# ABSTRACT

The Student Management System is a web-based application developed using the Django framework, designed to streamline administrative and academic activities in educational institutions. This system caters to three primary user roles: Admin, Staff, and Students. Each role is assigned specific functionalities, ensuring efficient management and accessibility of resources.

The Admin has full control over the system, including the management of users, subjects, courses, and other core data. Staff members can take attendance, manage results, respond to student feedback, and handle leave requests. Students can view their attendance records, exam results, apply for leave, and submit feedback.

A key highlight of this system is the integration of an AI-powered chatbot, which helps students navigate the platform and get instant responses to academic or administrative queries. Moreover, the system includes a questionanswer module for academic assessment, allowing staff to create quizzes or assignments and students to attempt them online.

This project aims to reduce manual workload, improve data accuracy, and enhance communication between students and faculty. By leveraging Django's secure, scalable architecture, the system ensures user-friendly interaction and role-based access control. The integration of AI and assessment tools makes the SMS not just a management tool but a smart academic companion for modern educational environments. at least 250 words.

Keywords - Django, Web Application , AI Chatbot , NLP , Machine learning , etc.

# **1. INTRODUCTION**

In the evolving landscape of digital education, managing student-related academic tasks efficiently has become increasingly important. This project introduces a robust **Student Management System (SMS)** developed using the Django framework, designed to cater to three primary user roles: **Admin, Staff, and Students**. The platform offers a wide range of functionalities such as **attendance monitoring**, **leave application management**, **feedback collection**, and **academic result publishing**, all through a centralized and user-friendly interface.

A key innovation of this system is the incorporation of **Machine Learning (ML)** and **Natural Language Processing (NLP)** techniques for **automated evaluation of subjective answers**. Traditional assessment methods for descriptive responses can be time-consuming and subjective. This system addresses those limitations by analyzing the context, grammar, keyword usage, and coherence of student-written answers to generate consistent and fair scores. This not only enhances the evaluation process but also saves considerable time for academic staff. Beyond assessment, the system includes an **AI-enabled chatbot** to assist students with common queries and navigation, enhancing accessibility and interaction. Furthermore, a **question-answer module** enables students to test their academic knowledge in a structured manner. By integrating AI and automation into conventional academic processes, this project aims to improve transparency, reduce administrative workload, and foster a more engaging learning environment.

# 2. PROBLEM STATEMENT

Traditional student management systems primarily focus on handling administrative tasks such as attendance tracking, result generation, and leave management. However, they often lack intelligent features to assist in **automated academic evaluations**, especially when it comes to **subjective answer assessment**. Evaluating descriptive responses manually is not only time-consuming but also prone to human bias and inconsistency. This becomes increasingly challenging in institutions with a high student-to-teacher ratio, where educators are burdened with large volumes of answer sheets.

Moreover, existing systems do not provide **personalized academic assistance**, leaving students with limited support outside the classroom. The absence of interactive tools such as **AI-powered chatbots** and **self-assessment modules** further restricts student engagement and academic growth.

Hence, there is a need for a **smart, automated, and scalable solution** that integrates the capabilities of Machine Learning and Natural Language Processing to evaluate subjective answers fairly, reduce educators' workload, and enhance student learning through intelligent assistance and self-assessment tools.

## **3. TECHNOLOGIES USED**

**Backend:** 

- Django: Python web framework for building the application.
- SQLite: Default database for development.
- OpenAI API: Used for the AI chatbot functionality.
- Firebase: For push notifications.

#### Frontend:

- HTML/CSS: For structuring and styling the web pages.
- Bootstrap: For responsive design and pre-built UI components.
- JavaScript: For dynamic content and AJAX requests.
- Chart.js: For displaying attendance statistics.

#### Libraries:

- requests: For making HTTP requests (e.g., Google reCAPTCHA validation).
- difflib: This is used to calculate the similarity between student answers and correct answers.
- openai: For integrating the OpenAI GPT-3.5-turbo model for the chatbot.

٠

## 4. FUNCTIONALITY AND WORKFLOW

#### Admin:

- 1. Dashboard:
  - View statistics (e.g., total students, staff, courses, subjects, attendance).
  - Upload and delete PDF documents for students.
- 2. Manage Staff/Students:
  - Add, edit, and delete staff and students.
  - Assign courses and sessions to students.
- 3. Manage Courses/Subjects/Sessions:
  - Add, edit, and delete courses, subjects, and sessions.
- 4. Notifications:
  - Send notifications to students and staff using Firebase.
- 5. View Feedback/Leave Applications:
  - View and respond to feedback and leave applications.

## <u>Staff:</u>

- 1. Dashboard:
  - View statistics (e.g., total students, attendance, leave applications).
- 2. Attendance Management:
  - Take and update attendance for students.
- 3. Result Management:
  - Upload and edit student results.
- 4. Question Management:
  - Create questions based on the dataset.
  - View student answers and similarity scores.
- 5. Feedback/Leave Applications:
  - Submit feedback and apply for leave.

## Students:

- 1. Dashboard:
  - View attendance statistics and course documents.
- 2. Attendance:
  - View attendance records for each subject.
- 3. Feedback/Leave Applications:
  - Submit feedback and apply for leave.
- 4. AI Chatbot:
  - Interact with the AI chatbot for academic assistance.
  - View chatbot conversation history.
- 5. Question-Answer Module:
  - View questions created by staff.
  - Submit answers and view results (PASS/FAIL).

# **5. SYSTEM ARCHITECTURE**

This system architecture outlines a comprehensive web-based academic management platform designed to cater to three distinct user roles: Admin, Staff, and Student. The workflow initiates at a Login screen where users authenticate their identities. Depending on their role, they are redirected to personalized homepages with specific functionalities and access controls tailored to their responsibilities.

#### **Admin Role**

For Administrators, the system directs them to the **Admin Homepage** post-login. The Admin has extensive control over the system and can manage the entire academic workflow. Their capabilities include:

- User Management: Admins can add, edit, or delete both staff and student records. This ensures smooth management of personnel data.
- **Course Management**: They have the authority to add, modify, or remove courses. This feature ensures the academic offerings are up-to-date and accurately represented.
- Attendance Management: Admins can view and track attendance across courses. They also have the ability to fetch attendance data, send detailed attendance reports, and manage leave applications from both staff and students.
- **File Management**: Admins can upload and delete files, such as PDFs, possibly containing course material, attendance reports, or other important academic documents.
- **Query Resolution**: Admins are responsible for answering queries, which could come from either staff or students, ensuring smooth communication within the platform.

#### Staff Role

Staff users, once authenticated, are directed to the **Staff Homepage**, which focuses primarily on academic data management and communication with students. Their responsibilities include:

- Attendance Management: Staff members can mark and track student attendance. They can retrieve student lists and input attendance data for courses they are assigned to.
- **Document Upload**: Staff have the option to upload relevant documents, such as PDFs containing attendance records or academic notes, which are then accessible to students or other authorized users.
- Query Handling: Staff are empowered to address student queries, providing academic support and clarifications when needed.

#### Student Role

The **Student Homepage** is designed to offer an intuitive interface for students to interact with the system. After logging in, students can:

- View Academic Information: Students can access their personal profile, attendance history, and academic results, providing them with a comprehensive view of their performance.
- Leave Management: Students can apply for leave, which is then reviewed and approved by the relevant authority (likely the Admin or Staff).
- **Feedback Submission**: Students can submit feedback regarding courses, faculty, or the overall academic experience, contributing to the improvement of the system.
- **Query Resolution**: Students can interact with the system's AI Assistant to ask questions or resolve queries. Additionally, they can send messages to staff or Admin to further address concerns.
- Notifications: Students receive real-time notifications about important updates, including attendance, upcoming exams, and leave approvals.

#### **Interactive Features**

To enhance the system's user experience, there are several interactive features:

- AI Assistant: This feature is designed to assist students with frequently asked questions, providing a more interactive and efficient way to resolve doubts. The AI assistant can answer academic-related queries, guide students through system processes, and provide instant help when needed.
- **Messaging System**: Both students and staff can send messages to one another, ensuring communication is streamlined and that all academic-related inquiries are addressed promptly.



**Fig – System Architecture** 

## 6. FLOWCHART

The flowchart illustrates a structured user authentication and navigation system designed for an educational institution or a similar organization. The process begins with a login step, where users must enter their credentials. If the credentials are invalid, users can reset their password. Once authenticated, the system directs users based on their role: Admin, Staff, or Student.

Admins have access to high-level administrative functions, including managing staff, students, and courses, as well as sending and viewing notifications. **Staff members** can perform tasks such as taking attendance, managing student results, applying for leave, giving feedback, and viewing their profiles. **Students**, on the other hand, can check their attendance, apply for leave, submit feedback, view their profiles, access academic results, review questions, and interact with an AI assistant for additional support.

The system ensures role-based access, with some overlapping features like leave applications and feedback submission available to both staff and students. The flowchart concludes once the user exits the system. This design enhances security and efficiency by providing tailored functionalities for each user type, streamlining institutional operations.



**Fig – Flowchart** 

# 7. UML

This UML diagram describes a system with three main features: **question answering, an AI chatbot, and PDF document sharing**. It's designed to support interactions between staff, students, and an automated assistant.

First, the **question-answering part** lets staff members create and list questions, which students can then view and answer. Think of it like a digital classroom where teachers post quizzes, and students submit their responses

Next, there's the **AI chatbot**—students can type messages, and the chatbot generates replies, acting as a virtual helper for answering questions or providing guidance.

Finally, the **PDF sharing feature** allows admins to upload documents (like lecture notes or study materials). The system keeps track of the latest uploads, and students can interact with these PDFs—perhaps by asking questions about the content, which the system then processes to generate relevant answers.

Overall, this setup blends human input with AI assistance, making it useful for education or any collaborative environment where information sharing and automated support are needed. The diagram keeps things structured but flexible enough for real-world use.



Fig – UML

## 8. CONCLUSIONS

This project is a robust and dynamic **Student Management System** designed to enhance the academic experience for administrators, faculty, and students alike. Built using cutting-edge technologies such as **Django** for the backend, **Firebase** for real-time database interactions and authentication, and **OpenAI** for intelligent communication features, the system ensures an intuitive and responsive interface across all user roles.

The platform is equipped with essential academic functionalities including **attendance tracking**, **feedback collection**, and **performance evaluation**, all integrated into a centralized portal. A standout feature is the **AI-powered chatbot** developed using OpenAI's language models, which assists students in resolving academic queries and navigating the platform efficiently.

One of the most innovative aspects of this system is its **question-answer evaluation module**, which utilizes **pre-trained datasets** and **similarity scoring algorithms** to evaluate student responses. This method provides more accurate and fair assessments by comparing answers against ideal solutions and calculating semantic similarity, moving beyond traditional keyword-based grading.

Additionally, the use of **data visualization and performance analytics** enables staff to monitor student progress in real time, while students receive actionable insights to improve their academic performance. The seamless integration of multiple technologies creates a holistic digital ecosystem for academic management and personalized learning support.

## 9. REFERENCES

[1] J. Wang and Y. Dong, "Measurement of text similarity: A survey," Information, vol. 11, no. 9, p. 421, Aug. 2020.

[2] M. Han, X. Zhang, X. Yuan, J. Jiang, W. Yun, and C. Gao, "A survey on the techniques, applications, and performance of short text semantic similarity," Concurrency Comput., Pract. Exper., vol. 33, no. 5, Mar. 2021.

[3] M. S. M. Patil and M. S. Patil, "Evaluating student descriptive answers using natural language processing," Int. J. Eng. Res. Technol., vol. 3, no. 3, pp. 1716–1718, 2014.

[4] P. Patil, S. Patil, V. Miniyar, and A. Bandal, "Subjective answer evaluation using machine learning," Int. J. Pure Appl. Math., vol. 118, no. 24, pp. 1–13, 2018.

[5] J. Muangprathub, S. Kajornkasirat, and A. Wanichsombat, "Document plagiarism detection using a new concept similarity in formal concept analysis," J. Appl. Math., vol. 2021, pp. 1–10, Mar. 2021.

[6] X. Hu and H. Xia, "Automated assessment system for subjective questions based on LSI," in Proc. 3rd Int. Symp. Intell. Inf. Technol. Secur. Informat., Apr. 2010, pp. 250–254.

[7] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in Proc. Int. Conf. Mach. Learn., 2015, pp. 957–966.

[8] C. Xia, T. He, W. Li, Z. Qin, and Z. Zou, "Similarity analysis of law documents based on Word2Vec," in Proc. IEEE 19th Int. Conf. Softw. Qual., Rel. Secur. Companion (QRS-C), Jul. 2019, pp. 354–357.

[9] H. Mittal and M. S. Devi, "Subjective evaluation: A comparison of several statistical techniques," Appl. Artif. Intell., vol. 32, no. 1, pp. 85–95, Jan. 2018.

[10] L. A. Cutrone and M. Chang, "Automarking: Automatic assessment of open questions," in Proc. 10th IEEE Int. Conf. Adv. Learn. Technol., Sousse, Tunisia, Jul. 2010, pp. 143–147.