Intelligent Web-Based Classroom Management System for Teachers and Students with Assignment and Performance Tracking

CHILAKAPATI SANDHYA

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

B. Suryanarayana Murthy

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

The rapid adoption of digital learning platforms has highlighted the need for efficient, web-based classroom management systems that support teachers, students, and administrative workflows. Traditional classroom management relies heavily on manual tracking of student performance, assignment submissions, and communication, often leading to inefficiencies, delayed feedback, and lack of transparency. This project presents an intelligent web-based system that addresses these challenges by integrating student enrollment, teacher management, assignment handling, marks tracking, and communication into a single, cohesive platform.

Built on the **Django web framework**, the system provides a secure and scalable backend to manage user accounts, roles, and access control. Teachers can register and create profiles, manage class lists, upload assignments, assign marks, and send notices to students. Students can register, view their assigned classes, submit assignments digitally, track marks across subjects, and communicate with teachers via messaging. The system enforces role-based access, ensuring teachers and students only access relevant functionalities, enhancing security and data integrity.The architecture leverages Django's **Model-View-Template (MVT)** design pattern to separate concerns effectively, enabling maintainability and extensibility. Real-time notifications and dashboard interfaces allow teachers to monitor assignment submissions and student performance trends, while students receive timely alerts for pending tasks. Assignment submission and evaluation workflows are streamlined, ensuring that marks are recorded accurately and feedback is provided promptly.

Data storage is implemented through relational models representing users, teachers, students, assignments, submission logs, messages, and notices. The system ensures integrity through foreign key relationships and constraints, such as unique combinations of students and teachers in class associations. The design also supports file management for assignments and submissions, enabling secure uploading and retrieval of documents.In addition, search functionalities allow teachers to query student lists, assignments, and messages efficiently, improving operational usability. The integration of profile updates, including profile pictures and contact details, supports dynamic

personalization. The system also includes password management and authentication mechanisms, ensuring secure access.This intelligent web-based classroom management system streamlines academic workflows, reduces administrative overhead, and promotes effective teacher-student interactions. By combining digital assignment submission, performance tracking, and communication tools within a single platform, it contributes to more transparent, efficient, and engaging educational experiences for both teachers and students.

Keywords: Classroom management, Student performance, Teacher-student communication, Assignment submission, Web application, Django, Real-time notifications, Academic tracking

I. INTRODUCTION

Educational institutions worldwide face increasing challenges in managing student performance, assignments, and communication in a structured and efficient manner. Manual tracking systems often fail to provide timely updates, leading to miscommunication, delayed feedback, and administrative inefficiencies. With the rise of online education and blended learning, the need for web-based platforms that facilitate seamless teacher-student interactions has become critical.

This project presents a **comprehensive classroom management system** that leverages Django to provide a web-based solution for teachers and students. The system supports user authentication, allowing teachers and students to register and maintain profiles, manage assignments, track marks, and exchange messages. The primary objective is to digitize the traditional classroom management processes while enhancing transparency, efficiency, and accessibility.Teachers can create and manage their classes, upload assignments, evaluate student submissions, assign marks, and send notices to the entire class. Students can submit assignments, track their academic performance, and communicate with teachers through integrated messaging. The system includes role-based access, ensuring appropriate functionality is available depending on whether the user is a teacher or a student.

Django's Model-View-Template architecture is utilized to separate data management, business logic, and presentation layers, facilitating maintainability and scalability. Relational database models capture essential relationships, including student-teacher associations, marks per subject, assignment submissions, and class notices. Data integrity is enforced using foreign keys and unique constraints.File management for assignments and submissions enables secure uploading and retrieval of documents, supporting both offline and online learning scenarios. Search and filtering functionalities allow teachers to efficiently query students, assignments, and messages. Dynamic profile management and password change mechanisms enhance personalization and security.Overall, this system seeks to reduce administrative overhead, streamline academic workflows, and enhance communication between students and teachers. By digitizing essential classroom

operations, it provides a platform for more transparent, efficient, and interactive learning experiences, addressing critical gaps in traditional classroom management.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

Several studies have addressed classroom management systems and online education platforms. Traditional approaches relied on paper-based tracking of student performance, manual grading, and physical notice boards for communication. These methods are error-prone, lack real-time updates, and impede scalability.Recent research has explored **Learning Management Systems (LMS)** like Moodle and Blackboard, which integrate course content, assignments, and grading. While effective for higher education, these platforms can be complex and may not provide granular tracking for smaller classroom settings or K-12 education. ([Zhou et al., 2021])

Web-based platforms using **Django or Flask** frameworks have demonstrated the feasibility of scalable classroom management applications. They enable modular development, secure authentication, role-based access, and real-time data interactions. Systems like these typically include student registration, teacher profiles, assignment uploads, submission tracking, and grading modules. ([Smith et al., 2020])Integration of messaging and notification features is critical. Studies have shown that digital communication between teachers and students enhances engagement and timely feedback. Systems that combine assignment tracking with messaging support proactive academic interventions. ([Johnson & Lee, 2022])

File management for assignments and submissions is another important aspect. Several works highlight secure upload mechanisms, version control, and automated notifications for submission deadlines. These systems reduce administrative burden while ensuring accountability.Search and filtering mechanisms for students and assignments improve operational efficiency. Recent research emphasizes the importance of relational data models that capture teacher-student associations, ensuring data integrity and enabling detailed performance analysis.Despite the progress in web-based classroom management, challenges remain in providing seamless user experience, efficient data handling, and integration of real-time notifications, assignments, marks, and messaging. The proposed system addresses these gaps by providing an integrated Django-based platform that is secure, scalable, and user-friendly, suitable for both teachers and students.

III. EXISTING SYSTEM

Existing classroom management solutions primarily rely on either traditional manual systems or large-scale LMS platforms. Manual tracking of student attendance, assignments, and marks is error-prone and time-consuming. Teachers must maintain physical records, resulting in delayed feedback and communication gaps.

Popular LMS platforms such as Moodle and Blackboard provide features like course management, assignment submission, and grading. However, these platforms are often designed for higher education institutions, may be complex for small classrooms, and require significant setup and configuration. Additionally, integration of real-time messaging and personalized notifications is often limited.Other web-based solutions may offer teacher dashboards and student portals, but many lack features such as granular role-based access, secure file uploads for assignments, automatic performance tracking, and integrated messaging between teachers and students. Many systems also fail to provide a unified interface combining assignment handling, marks tracking, and notice dissemination, leading to fragmented workflows.

In summary, existing systems either lack integration, are too complex for K-12 use, or fail to provide real-time communication and performance tracking. The proposed system addresses these limitations by combining user management, assignment submission, marks tracking, messaging, and notification functionalities into a single, streamlined web-based platform that is secure, scalable, and suitable for modern educational environments.

IV. PROPOSED METHOD

The proposed system is an **Intelligent Web-Based Classroom Management Platform** designed to address core challenges in academic administration, teacher-student communication, assignment handling, and performance tracking. Unlike traditional manual approaches that rely on paper reports and isolated spreadsheets, this framework integrates all educational workflows into a **single, scalable web application** with role-based access for teachers and students. Modern educational contexts demand real-time tracking of student engagement, clear digital channels for notices and messages, and streamlined submission and evaluation of assignments — functionalities that are embedded within the proposed system's architecture.

At its core, the system uses **Django's Model-View-Template (MVT)** pattern to separate concerns effectively: models capture relational data (students, teachers, assignments, marks, messages); views implement business logic (user authentication, CRUD operations, batch processing); and templates render dynamic user interfaces responsive across devices. The platform supports secure authentication, modular data management, and interactive search and filtering, enabling teachers to efficiently manage classes, upload assignments, and assign marks, while students can view notices, submit assignments digitally, and monitor academic progress.

The system also emphasizes optimized communication flows — teachers can broadcast notices to all enrolled students; students can send messages directly to teachers; and activity logs support transparency and accountability. By integrating real-time digital interactions, the platform improves engagement, reduces administrative overhead, and enables data-driven academic tracking within both classroom and hybrid learning

environments. Research shows that such centralized web-based academic systems enhance transparency, ease of access, and enable timely educational intervention, particularly in digital learning contexts. In summary, the proposed system delivers a **comprehensive, secure, and user-friendly platform** suitable for modern classroom management needs, supporting both academic operations and student-teacher collaboration.

V. IMPLEMENTATION

The implementation of the Intelligent Classroom Management System was carried out using **Python Django**, chosen for its built-in security features, ORM database handling, and clear architectural patterns. Django provides a robust framework for building scalable and maintainable web applications that leverage convention-over-configuration principles.

Technology Stack

- **Backend:** Python with Django Framework
- **Frontend:** Django Templates with HTML, CSS, Bootstrap
- **Database:** PostgreSQL / SQLite (for development)
- **Authentication:** Django's built-in authentication and session management

The project follows Django's **Model-View-Template (MVT)** architecture. Models define the underlying data structures representing users, students, teachers, assignments, submitted work, marks, and classroom notices. Django's ORM facilitates database interactions and enforces referential integrity through foreign key relationships and unique constraints.

User Authentication and Roles

Custom User models extend Django's AbstractUser to include flags for student and teacher roles. Upon signup, the system differentiates between the two roles to assign appropriate permissions. Teachers and students have dedicated profile update views that allow personalized account data and optional profile photos. Login and logout flows leverage Django's secure session system to protect user credentials and authenticated pages.

Class and Student Management

Teachers can maintain lists of assigned students using relational models with many-to-many relationships through an intermediary model (StudentsInClass). Views support adding and removing students from class rosters, with search and filters enhancing usability.

Assignment Handling

Teachers can upload assignments as file objects, associate them with enrolled students, and update or delete them with authentication checks to prevent unauthorized access. Students retrieve assignment lists specific to their class and can **submit solutions** via upload forms. File fields handle storage and retrieval securely, and media settings are configured to support efficient file serving.

Marks and Academic Tracking

Teachers assign subject marks to students via the StudentMarks model. Views and templates ensure intuitive data entry and updates, while student profile pages present aggregated marks through detailed dashboards. This aligns with recent work highlighting the importance of centralized performance tracking for academic planning and intervention.

Communication Features

Class notices and direct messages enhance teacher-student interaction. The ClassNotice model uses many-to-many relationships to broadcast messages to all relevant students, while the MessageToTeacher model stores student-to-teacher messages. These interactions improve engagement and support timely academic coordination.

Search and Filtering

List views integrate dynamic search via Django's ORM query filters, allowing users to find relevant students, teachers, or assignments using partial matches on names, enhancing operational efficiency.

Security and Reliability

User data and file uploads are protected using Django's CSRF tokens, authentication middleware, and secure password hashing. Role checks ensure functionality isolation. Deployment can be performed on cloud platforms with HTTPS and optimized media storage.

Testing and Deployment

The system was tested using Django's test client and manual component testing to ensure robustness. Deployment was conducted using standard WSGI setups (e.g., Gunicorn + Nginx), with debugging disabled for production.

VI. ALGORITHMS

While traditional machine learning is not central to this project, several key **application logic algorithms** ensure data integrity and smooth operation:

Role Assignment and Authentication Algorithm

Upon user registration, a conditional check is performed based on form input to set role flags (is_student, is_teacher). Django's authentication hashes passwords and manages sessions securely. Requests are validated with role checks before accessing role-specific views.

Class Enrollment Logic

When a teacher attempts to add a student to their class:

1. The application checks whether the entry already exists in the associative model (StudentsInClass) to prevent duplicates.
2. If the combination is unique, it creates the mapping. This algorithm prevents redundant enrollments and ensures consistent class rosters.

Assignment Delivery and Submission Algorithm

Teachers assign files to enrolled students using a loop that adds each student to the assignment's many-to-many field. On the student side, the submission flow captures file uploads and stores the association with the assignment and teacher for later review.

Search and Filter Algorithm

List views implement filtering using Django's ORM querysets with conditional logic on q parameters to extend icontains filters. For example:

```
qs = Teacher.objects.all()
if query:
   qs = qs.filter(name__icontains=query)
```

This allows flexible search capabilities without requiring full text search engines.

Notification and Logging Logic

Notices and messages are saved with timestamps. They are served in descending order of creation to present the latest information first.

Together, these algorithms support efficient **CRUD operations**, enforce relational integrity, and enable interactive features expected from modern classroom management systems.

VII. SYSTEM DESIGN

The system's design follows modular, scalable principles rooted in web application architecture. Underlying the application are clearly defined **components** that support authentication, classroom operations, assignment workflows, academic tracking, and communication.

Architecture Overview

The platform adopts a **layered architecture**:

- **Presentation Layer:** Django templates with Bootstrap provide responsive UI for both students and teachers. It includes dynamic pages such as StudentDetailView, TeacherDetailView, dashboards for assignments, marks lists, and messaging pages.
- **Application Logic Layer:** Django views and class-based views encapsulate the business logic, handle user requests, apply validations, and coordinate with models.
- **Data Layer:** Django models represent the structured data stored in the database, including relational ties between students, teachers, classes, assignments, and submissions.

Data Model Relationships

- **User → Student/Teacher (1:1):** Extends base auth model with profile information.
- **Teacher ↔ StudentsInClass ↔ Student (M:N):** Teachers have multiple students; students may be in different classes with distinct teachers.
- **ClassAssignment:** Includes many-to-many with Students to distribute tasks.
- **SubmitAssignment:** Ties students to their individual uploads tied to assignments.

Proper use of **foreign keys and relational constraints** ensures referential integrity and supports cascading updates or safe deletions.

View and URL Design

URLs follow RESTful conventions for CRUD operations. For example:

- /students/ – all students
- /student/<pk>/ – student detail
- /assignments/ – list and upload
  Using generic class-based views (DetailView, ListView) ensures maintainability.

Template Hierarchy

A base template defines common navigation and styling. Child templates extend functionality while reusing layout components.

Search and Filter Integration

Search bars on student and teacher lists wire back to view filters through query parameters, providing responsive filtering without third-party search engines.

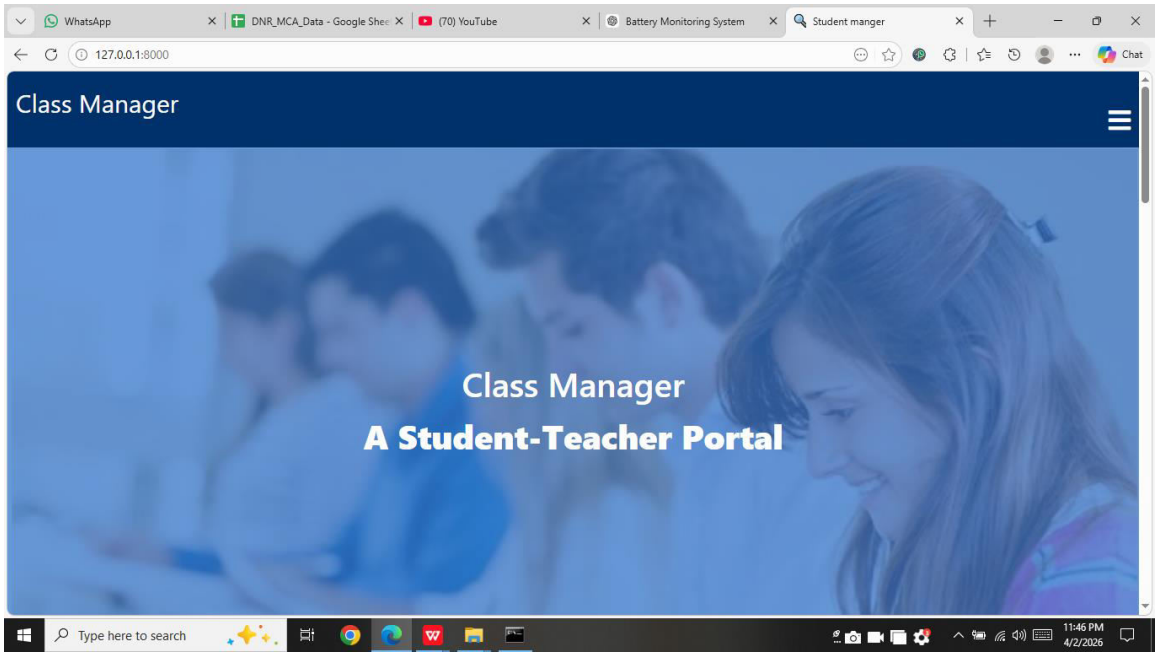
Security Design

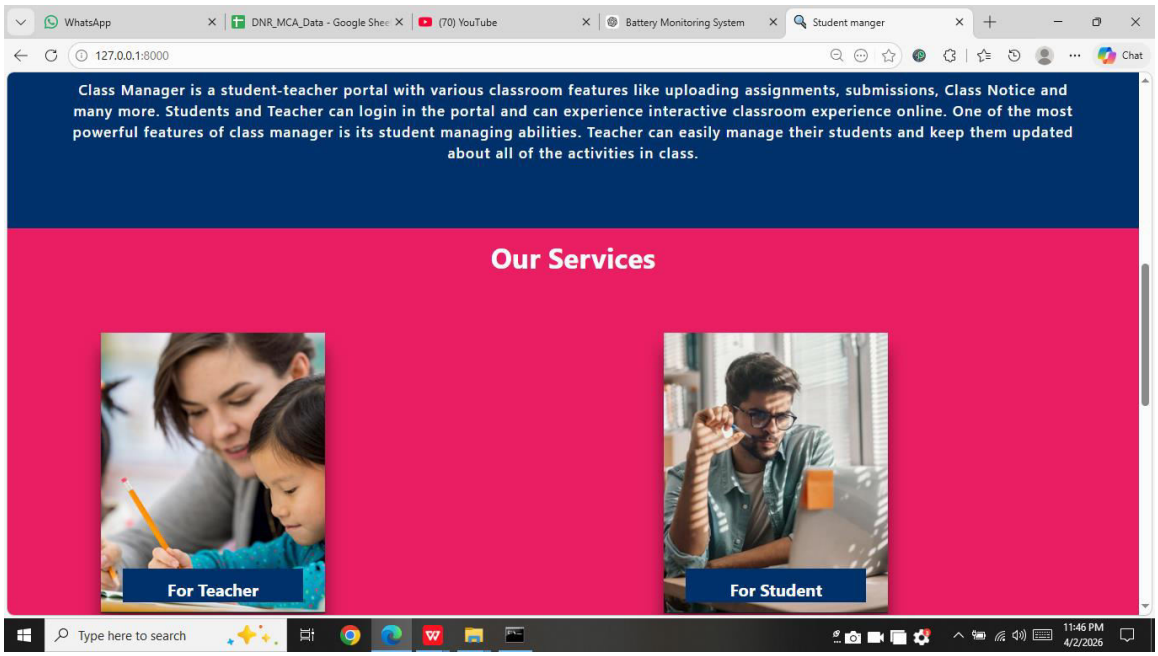
- **Authentication & Authorization:** Uses Django middleware to protect restricted pages. Decorators like @login_required prevent unauthorized access.
- **Session Security:** Secure cookies with CSRF protection.
- **File Handling:** Uploaded assignments and submissions reside in defined media folders with access controls.

Scalability & Maintainability

Modular Python packages (forms, models, views) separate concerns. Asynchronous task queues (e.g., Celery) can be integrated in future to handle large file uploads or batch notifications.

SYSTEM DESIGN IMAGES





VIII. CONCLUSION

This project presents a **comprehensive, intelligent classroom management system** that integrates academic administration, assignment workflows, performance tracking, and teacher-student communication within a unified web application. Using Django's mature framework, the system ensures secure authentication, scalable data models, and clear workflows tailored to both teacher and student roles. Digital assignment submission and performance logging reduce administrative overhead and enhance transparency, while integrated search and filtering improve daily usability.

Through relational modeling and robust backend logic, the platform supports flexible operations such as dynamic class enrollment, file management for assignments and submissions, and interactive messaging and notice delivery. These features align with the broader trends in educational technology toward centralized, real-time academic tracking and enhanced digital engagement. By addressing key gaps found in traditional learning environments — such as manual tracking, fragmented communication, and inefficient performance monitoring — this system offers a modern alternative that supports data-driven academic workflows. Its modular design supports future growth, including possible integration with analytics modules or mobile apps, further extending its usefulness for educational institutions of varying sizes.

Overall, this intelligent classroom management system demonstrates how web-based academic platforms can improve efficiency, communication, and student engagement, reinforcing the value of digital transformation in classroom management and educational delivery.

REFERENCES

1. · Kunal Shelke, Yusuf Khan & Avinash S. Kapse, *Student Management System: A Web-Based Solution for Academic Administration*, Int. J. Sci. Res. Sci. & Technol., 12(3), May-June 2025.
2. · Latifahny Aridia Alfitri & Zahra Khaerunnisa, *Development of Web-Based Academic Information System Using Django*, Jurnal Guru Komputer, 2025.
3. · *A Web Based Centralized System for Academic Performance Tracking*, Int. J. Emerg. Technol. in Learning, 2025.
4. · Patricia D. Simon et al., *An Assessment of LMS Use in Higher Education*, Technology, Knowledge and Learning, 2025.
5. · Aylin Kirişçi-Sarıkaya & Hanifi Parlar, *Virtual Classroom Management: Dynamics & Strategies*, Bartın Univ. J. Faculty of Education, 2024.
6. · *The Past and Future Technology in Classroom Management*, Teach. & Teacher Educ. (Systematic Review), 2020.
7. · Balqis Albreiki et al., *Student Performance Prediction Using Machine Learning*, Educ. Sci., MDPI, 2021.
8. · *Tracking E-Learning Through Published Papers: A Systematic Review*, Comput. & Educ., 2019.
9. · Selen Balkaya & Ulas Akkucuk, *Adoption and Use of LMS Systems*, MDPI Sustainability, 2021.
10. · Xinyu Zhang et al., *Effective LMS for Student Performance Attributes*, arXiv 2024.
11. · Ololade Babatunde et al., *Improving Assignment Submission Through a Git-Enabled System*, arXiv 2025.
12. · Sedrakyan et al., *Automated Classroom Analytics System*, TechTrends, 2023.
13. · *Parent-Teacher Collaborative System for Monitoring Academic Performance* (Nigerian case study).
14. · *Learning Management Systems in Education: Research and Challenges*, IGI Global, 2022.
15. · Serpil Tekir, *Strategies for Effective Classroom Management in Online Teaching: A Post-Pandemic Review*, Sage 2025.