WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Timetable Schedular and Monitoring System

Sujatha N[1], Monica M[2], Madhu T[3], Monica P[4], Mahashree A M[5]

[1,2,3,4,5] *dept.of Artificial Intelligence and Machine Learning, Sri Shakthi Institute of Engineering and Technology (Affiliated to Anna University)* Coimbatore, India

madhut23aml@srishakthi.ac.in, monicam23aml@srishakthi.ac.in, madhut23aml@srishakthi.ac.in, monicap23aml@srishakthi.ac.in

mahashreeam23aml@srishakthi.ac.in

Abstract—

This project presents a web-based faculty management system designed to streamline class substitution in educational institutions. The system enables staff to log in securely and manage their profiles and schedules. When a faculty member is marked absent, the application automatically checks for available substitute staff without timetable conflicts and assigns them accordingly. Real-time notifications are sent via email to inform the substitute professor, including time-specific greetings (e.g., "Good Morning" or "Good Afternoon"). Additionally, the system offers timetable viewing and management functionality through an intuitive interface. Developed using Flask for the backend, SQLite for database management, and HTML/CSS for the frontend, the platform enhances communication, minimizes class disruptions, and supports seamless substitution management. This project demonstrates the integration of scheduling logic with real-time alert mechanisms to optimize institutional workflow.

Keywords:- Faculty Substitution, Email Alert, Real time Scheduling, Substitute Allocation

1. Introduction

Managing faculty schedules and ensuring smooth class coverage during staff absences is a recurring challenge in educational institutions. Manual substitution processes often lead to confusion, communication delays, and disrupted learning experiences. This project introduces a web-based faculty substitution and attendance management system that automates key tasks such as login authentication, timetable display, absence marking, and substitute assignment. When a faculty member is marked absent, the system automatically identifies an available staff member whose schedule does not conflict and sends them a timely notification via email. The platform also includes time-based greetings and a user-friendly interface for staff to access and manage their timetables. Developed using Flask for backend processing, HTML/CSS for frontend design, and SQLite for data storage, the system aims to improve operational efficiency, reduce administrative workload, and maintain the continuity of teaching activities.

2. LITERATURE REVIEW

A. Existing Systems

Traditional faculty management systems in educational institutions typically rely on manual record-keeping, which can result in scheduling conflicts and inefficient substitution during faculty absences. Although some institutions use ERP systems or basic scheduling software, these tools often lack real-time substitution features and require significant administrative oversight. Studies have shown that while these systems can manage attendance and class schedules, they do not dynamically reassign duties or notify substitutes when a faculty member is unavailable. This leads to missed classes and disrupted learning processes.

B. Technological Advancements and Proposed System

Recent developments in web technologies and automation have enabled more robust solutions for academic scheduling. Web frameworks like Flask, along with databases such as SQLite or MySQL, allow for the creation of responsive, scalable applications. Research in automated timetable generation and notification systems supports the use of AI or logic-based substitution allocation to optimize resources. The proposed system builds on these advancements by offering automatic faculty substitution based on real-time availability, preventing timetable clashes, and sending timely email notifications with personalized greetings. This ensures smoother academic operations and minimizes manual effort.

3. METHODOLOGY

A. Requirement Analysis

The first step in developing the faculty substitution system involves understanding the needs of the users and the problems they face. The primary users include faculty members and administrators. Requirements include login and registration functionality, the ability to manage and display timetables, mark absences, find substitutes without schedule conflicts, and send timely email notifications. An emphasis is placed on automation to reduce manual effort and errors in assigning substitutes.

B. System Design

Based on the requirements, system architecture and workflow are designed. The database is modeled using an Entity-Relationship Diagram (ERD), showing relationships between entities like Users, Timetables, Attendance, and Substitution. Web interfaces are sketched using wireframes to provide intuitive navigation for users. A substitution flowchart is created to ensure the logic is clear: when a faculty is marked absent, the system searches for a free faculty at that hour and assigns them automatically.

C. Database Setup

A lightweight database like SQLite is used to store user and timetable data. Tables such as Users, Timetable, Attendance, and Substitution are created. The Users table stores login credentials, while the Timetable table holds data about class schedules. The Attendance table records absences, and the Substitution table logs who substituted whom and when. Relationships between tables are enforced with foreign keys to maintain data integrity and avoid redundancy.

D. Backend Development Using Flask

The Flask framework is used to build the backend. Secure routes are created for login, registration, and dashboard access. Business logic includes identifying absent faculty, checking timetable conflicts, and assigning substitutes. The backend queries the database to ensure that substitute staff do not already have a class during that time slot. Once assigned, the system records the substitution and triggers a notification to the new assigned faculty member.

E. Frontend Development

The frontend is developed using HTML, CSS, and JavaScript. The login and registration pages include input validation, while the dashboard displays staff names and allows selection to view individual timetables. The admin interface includes features to mark staff absent and view automatic substitution results. JavaScript ensures form validation, while CSS makes the interface responsive and user-friendly.

F. Notification System Integration

To alert substitute faculty, an email notification system is integrated using Python's smtplib. When a staff member is assigned as a substitute, they receive a real-time email. The system dynamically adjusts the message to include a greeting such as "Good Morning" or "Good Afternoon," based on the time of day. This ensures professional and timely communication without manual intervention.

G. Testing and Validation

Before deployment, the system undergoes thorough testing. Unit tests check the correctness of individual components, such as login validation and substitution logic. Integration testing ensures modules work together seamlessly. User acceptance testing is conducted with a small group of faculty and administrators to verify usability and correctness under real-world scenarios.

4. CONCLUSION

This methodology outlines a complete roadmap for implementing a faculty substitution and notification system as a web application. By combining a robust backend, a user-friendly frontend, automated substitution logic, and real-time communication, the system addresses key challenges in educational scheduling. It improves institutional efficiency, reduces manual workload, and ensures uninterrupted class management through timely substitutions and notifications.

Acknowledgment

I sincerely thank my project guide Sujatha for their valuable guidance and support throughout this project. I also express my gratitude to the faculty and staff of the Artificial Intelligence and Machine Learning for their encouragement. Special thanks to my project teammates for their collaboration and dedication, and to my friends and family for their constant support and motivation during this journey.

REFERENCES

[1] A Smart Approach to Dynamic Faculty Substitution A. Jain and M. Gupta (2018)

[2] Al-Based Timetable Scheduling System for Educational Institutions S. Rani and V. Prasad (2020)

[3] Automated Timetable Generation using Genetic Algorithm P. Tripathy, R. Patra, and S. Mishra (2019)

[4] Design and Implementation of a Web-Based Timetable Management System M. Alkan and S. Turkoğlu (2017)

[5] Smart Class Substitution Management System Using Android Application A. Dhamodharan and P. Nivetha (2021)

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an MSW document, this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord "Format" pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.