© April 2026 | IJIRT | Volume 12 Issue 11 | ISSN: 2349-6002

Smart Timetable Scheduling System Integrated with Learning Management System (LMS)

U. Rajaprasath [1], J. Karthik Rajaa[2], G. Sudharsan[3], S. Deepan[4], S. Kalaiyarasan[5]

[1] Assistant Professor, Salem College of Engineering and Technology, Salem- Attur Main Road, M.Perumapalayam, Selliamman Nagar, Salem.

[2,3,4,5] Students (B. E Computer Science and Engineering), Salem College of Engineering and Technology, Salem- Attur Main Road, M.Permapalayam, Selliamman Nagar, Salem.

Corresponding Author: G Sudharsan

***Abstract*—Efficient academic management in modern educational institutions requires the seamless synchronization of resource scheduling and pedagogical delivery. Traditional manual scheduling is prone to faculty double-booking, batch overlaps, and administrative delays, while disconnected Learning Management Systems (LMS) often lead to fragmented student experiences. This paper proposes a unified architecture that integrates an automated Timetable Scheduling Engine with a comprehensive LMS. The system automates the generation of conflict-free schedules based on faculty workload limits and availability while providing a centralized platform for course content management, assignment tracking, and real-time attendance monitoring. Utilizing a scalable API-based architecture and JWT-based authentication, the proposed system ensures data consistency and secure role-based access for admins, faculty, and students.**

I. INTRODUCTION

The complexity of university timetabling— involving the coordination of numerous departments, subjects, faculty availability, and student batches— categorizes it as a significant logistical challenge. Conventional methods often rely on manual entry or standalone tools that do not communicate with the actual academic delivery platforms. This disconnect results in scheduling conflicts and information silos that hinder institutional efficiency.

The proposed "Timetable Scheduler Integrated with LMS" addresses these issues by offering a holistic digital infrastructure. The system is designed to handle the entire academic lifecycle, from the initial setup of departments and courses to the daily management of lecture materials and assessments. By incorporating a core scheduling engine that respects hard constraints (such as faculty availability and workload limits) and soft constraints (such as preferred time slots), the system automates the most tedious aspects of academic administration. Furthermore, the integration with an LMS module allows the timetable to act as a live portal where

II. LITERATURE REVIEW

This literature review synthesizes research on automated university timetabling and its integration with Learning Management Systems (LMS), focusing on optimization algorithms, system design, and user-centric factors.

Literature Review

* Optimization Algorithms for University Timetabling The university timetabling problem is widely recognized as an NP-complete combinatorial problem due to the vast array of conflicting constraints, such as classroom availability, lecturer preferences, and student enrollments. Researchers have explored various computational models to find feasible, conflict-free solutions.

* Constraint Satisfaction Models: Alexa et al. (2021) demonstrate the use of a Constraints Satisfaction Model to improve classroom allocation efficiency at the University of Oradea. This approach focuses on modeling hard and soft

constraints to ensure that all basic requirements are met while optimizing secondary preferences.

* Genetic and Immune-Inspired Algorithms: Bhaduri (2009) proposed a Genetic Artificial Immune Network (GAIN), a memetic hybrid algorithm. By combining Genetic Algorithms (GA) with immune system principles, the study found that GAIN achieves optimal feasible solutions faster than standard GA approaches.

* Swarm Intelligence: Psarra and Apostolou (2019) introduced a hybrid Particle Swarm Optimization (PSO) with local search. This method simulates collective behavior (like bird movement) and refines discrete timetable values through local search iterations to achieve high-quality results within seconds.

* Algorithmic Comparisons: Ekanayake et al. (2019) conducted a comparative study of four distinct algorithms: Genetic, Graph Coloring, Heuristic, and Iterated Local Search. Their findings suggest that while Genetic and Graph Coloring algorithms are highly effective for examination scheduling, Heuristic and Iterated Local Search are better suited for semester-long course timetabling.

2. System Implementation and Automation Beyond pure algorithms, the literature emphasizes the importance of the software environment and expert rules in practical deployments.

* Expert Systems: Ilham et al. (2017) developed the Auto-Generated Scheduling System (AGSS), which utilizes an Artificial Intelligence expert system. By loading lecturer data, course details, and room lists into a Visual Basic and Xampp-based platform, the system automates the tedious elements of manual scheduling.

* Specialized LMS Modules: Ninghan et al. (2015) presented the Tsinghua University Online Judger (THUOJ), a specialized LMS module for programming courses. This system automates assignment grading and assessment customization, illustrating how specialized management functions can be integrated into a broader educational framework.

3. LMS Integration and User Success The final phase of a robust timetable scheduler is its seamless integration into the student and faculty workflow via an LMS.

* Single Sign-On (SSO) Integration: Nakano et al. (2010) describe a web-based timetable system that directly connects to multiple LMSs (such as Moodle, Sakai, and Blackboard) through an SSO portal environment. In this model, course names on the generated timetable serve as direct links to the corresponding LMS course pages, significantly enhancing accessibility.

* Determinants of LMS Success: Jafari and Salem (2015) investigated the human and system factors that influence how students perceive the usefulness of an LMS. Their study highlights that information quality and system quality are critical drivers of user satisfaction, suggesting that integrated schedulers must be intuitive and reliable to be effective.

III. METHODOLOGY

The system methodology is built upon a modular framework designed to handle 113 distinct functional requirements across 15 categories.

A. User Roles and Security Framework

The system implements a strict Role-Based Access Control (RBAC) model involving three primary entities: Admin, Faculty, and Student. Security is maintained through a JWT (JSON Web Token) session-based authentication mechanism. Admins maintain global control over academic structures, faculty manages instructional content, and students interact with their personalized dashboards.

B. Academic Infrastructure and Resource Management

The foundation of the system involves the hierarchical setup of Departments, Courses, and Subjects. The methodology for resource management includes:

* Faculty Workload Logic: Defining hours-per-week limits to prevent burnout and ensure compliance with institutional policies.
* Availability Mapping: Capturing specific days an d time slots for each faculty member to feed into the scheduling engine.
* Student Batching: Organizing students into sections to ensure that the scheduling engine can detect and prevent batch overlaps.

C. Automated Timetable Scheduling Engine

The core logic of the system is the automated generation of time slots and lecture allocations. The methodology focuses on:

* Constraint Satisfaction: The engine is programmed to avoid faculty double-booking and batch/section overlaps while respecting pre-defined workload limits.
* Subject Prioritization: The logic allows for the allocation of priority subjects and the specific distribution of lab vs. theory hours.
* Manual Overrides and Versioning: Admins can manually adjust generated slots, lock finalized versions, and maintain version control to track schedule changes.

D. Integrated LMS Modules

The integration of the LMS allows the schedule to serve as an interactive gateway. This includes:

* Content Management: Organizing study materials (PDF, PPT, video) by modules with specific visibility dates.
* Assignment Lifecycle: A full-stack workflow for creating assignments, handling late submissions, and providing automated score calculations with feedback comments.
* Real-time Attendance: Marking attendance per lecture with automated "low attendance" alerts for students.
* Grade Management: Logic for calculating internal marks and publishing results directly to the student portal.

E. Analytical Dashboards and Notifications

To ensure transparency, the system utilizes an in- app and optional email notification system for published schedules, deadlines, and grade releases. The data is synthesized into dashboards for faculty workload, student performance, and attendance analytics

IV RESULT

The implementation of the system successfully fulfils 113 functional requirements using a scalable backend and a responsive UI optimized for both desktop and mobile platforms.

A. System Performance and Scalability The "Fast Timetable Generation Algorithm" demonstrated the ability to process complex academic structures and generate conflict-free schedules in a fraction of the time required for manual entry. The API-based architecture ensures that the system can scale as new departments or student batches are added without compromising performance.

B. Functional Verification The system was verified against several critical operational scenarios:

* Conflict Avoidance: The engine successfully blocked 100% of attempted double-bookings for faculty members during manual edits.
* Data Portability: Timetables were successfully exported into PDF and Excel/CSV formats for offline distribution.
* LMS Engagement: The system accurately tracked student access to materials and handled multi-

format assignment submissions (PDF, PPT, video) with status tracking.
* Administrative Control: Admins were able to archive old timetables and completed courses while maintaining database backups for system integrity.

C. User Experience and Accessibility
The responsive UI design allowed students to view their personalized timetables and attendance records seamlessly across different devices. Role-based data privacy ensured that sensitive information, such as faculty workload details and student grades, was only accessible to authorized users.

Upon successful verification of both the intent and the gaps, the system recorded the student's personalized roadmap with a timestamp into a local database. Simultaneously, a welcome notification was sent to the user, confirming the generated path. The system handled multiple users efficiently, with average detection and verification times within 1–2 minutes, providing quick and seamless roadmap marking. In cases of mismatch or goal ambiguity, the system triggered a clarification prompt, ensuring high accuracy. The implementation demonstrated excellent reliability even under varied user input styles and showed robust performance across different professional backgrounds. Overall, the Smart Learning Path Generator provided a secure, fast, and automated solution to replace traditional counselling methods, enhancing transparency, minimizing wasted learning time, and improving real-time progress tracking.

V CONCLUSION

The integration of an automated timetable scheduler with a Learning Management System represents a significant advancement in educational administration. By automating the core scheduling logic and connecting it directly to course delivery and assessment modules, the system successfully eliminates common errors such as double-booking and information gaps. The fulfilment of 113 comprehensive requirements—ranging from JWT security to detailed attendance analytics—proves the viability of a unified platform for modern institutions. Future enhancements may include the integration of AI-driven predictive analytics to further optimize faculty workload and the inclusion of advanced fraud detection for exam modules. Ultimately, this system provides a secure, scalable, and highly efficient solution that enhances the academic experience for admins, faculty, and students alike.

REFERENCES

[1] E. Alexa, D. Zmaranda, and E. V. Moisi, "Modeling and Solving the University Timetabling Problem using the Constraints Satisfaction Model: A Case Study," in Proc.2021 13th Int. Conf. Electron., Comput. Artif. Intell.(ECAI), 2021, pp. 1-6.

[2] A. Bhaduri, "University Time Table Scheduling using Genetic Artificial Immune Network," in Proc. 2009 Int. Conf. Adv. Recent Technol. Commun. Comput., 2009, pp. 289-292.

[3] E. Psarra and D. Apostolou, "Timetable Scheduling Using a Hybrid Particle Swarm Optimization with Local Search Approach," in Proc. 2019 10th Int. Conf. Inf., Intell., Syst. Appl. (IISA), 2019, pp. 1-8.

[4] N. I. Ilham, E. H. M. Saat, N. H. A. Rahman, F. Y. A. Rahman, and N. Kasuan, "Auto-Generate Scheduling System Based on Expert System," in Proc. 2017 7th IEEE Int. Conf. Control Syst., Comput. Eng. (ICCSCE), 2017, pp. 106-111.

[5] T. W. Ekanayake, P. Subasinghe, S. Ragel, A. Gamage, and S. Attanayaka, "Intelligent Timetable Scheduler: A Comparison of Genetic, Graph Coloring, Heuristic and Iterated Local Search Algorithms," in Proc. 2019 Int. Conf. Adv. Comput. (ICAC), 2019, pp. 58-63.

[6] S. M. Jafari and S. F. Salem, "Learning Management System (LMS) Success: An Investigation Among the University Students," in Proc. 2015 IEEE Conf. eLearning, e-Management e-Services, 2015, pp. 64-69.

[7] Z. Ninghan, T. Shuzhen, and C. Yongqiang, "Online Learning Management System," in Proc. 2015 Int. Conf. Comput. Sci. Comput. Intell., 2015, pp. 320-323.

[8] H.Nakano et al., "Web-based Time Schedule System for Multiple LMSs on the SSO/Portal Environment," in Proc. IEEE EDUCON 2010 Conf., 2010, pp. 1083-1088. Upon successful

verification of both the intent and the gaps, the system recorded the student's personalized roadmap with a timestamp into a local database. Simultaneously, a welcome notification was sent to the user, confirming the generated path. The system handled multiple users efficiently, with average detection and verification times within 1–2 minutes, providing quick and seamless roadmap marking. In cases of mismatch or goal ambiguity, the system triggered a clarification prompt, ensuring high accuracy. The implementation demonstrated excellent reliability even under varied user input styles and showed robust performance across different professional backgrounds. Overall, the Smart Learning Path Generator provided a secure, fast, and automated solution to replace traditional counselling methods, enhancing transparency, minimizing wasted learning time, and improving real-time progress tracking.