International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

Digital Transformation and Emerging Technologies Sarms: Smart Academic Resource Management System

K S SHRAVANI and MEGHANA S
Department of Computer Applications, B.M.S. College of Commerce and Management, Bengaluru, Karnataka, India
Project Guide: Asst PROF BALARAM S.
Email: meghana227s@gmail.com
Corresponding author: shravanikatta24@gmail.com

TABLE OF CONTENTS

1. Abstract
2. Keywords
3. Introduction
4. Literature Review
5. Research Gap
6. Problem Statement
7. Aim and Objectives
8. Research Questions
9. Methodology
10. Proposed System and Architecture
11. Key Findings / Results
12. Discussion
13. Limitations
14. Future Scope
15. Conclusion
16. References

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

1. Abstract - Preparing a college timetable involves coordinating classes, faculty, rooms, laboratories, language groups and shared subjects without conflicts. This paper presents SARMS (Smart Academic Resource Management System), a web-based system for academic resource management and timetable generation. It uses React and Vite for the interface, Node.js and Express.js for the backend, and MySQL/MariaDB for data management. The generator follows a priority-based approach, scheduling language groups and shared subjects first, followed by laboratories and regular lectures. It checks class, faculty and room availability and selects the smallest suitable room. A validation stage detects clashes, capacity issues and room suitability problems. SARMS does not claim a new optimisation algorithm; its contribution is integrating institution-specific scheduling rules into a practical digital system with diagnostic reporting for unplaced sessions. This supports Digital Transformation by replacing fragmented manual scheduling with a centralised workflow. Further testing is required to evaluate generation time, completeness and improvement over manual practices.

Keywords: Academic resource management; university course timetabling; classroom allocation; constraint validation; digital transformation.

2. Introduction

College timetable creation is a complex resource-allocation problem involving classes, faculty, rooms, laboratories and academic constraints. Teachers and rooms cannot be double-booked, while rooms must meet capacity and facility requirements. Language groups, shared subjects and multiple sections further increase complexity. University timetabling is therefore considered a difficult combinatorial problem (Babaei, Karimpour and Hadidi, 2015; Chen et al., 2021).

Existing approaches include mathematical programming, heuristics and metaheuristics, but institutional requirements often differ from standard benchmark problems. Research also highlights a gap between timetabling models and practical institutional software (Oude Vrielink et al., 2019).

SARMS addresses this gap through a web-based system that integrates college-specific scheduling rules, resource validation and diagnostic reporting. It prioritises restrictive activities and manages limited classrooms, laboratories and faculty resources through a centralised digital workflow, supporting the conference theme of Digital Transformation and Emerging Technologies.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

3. Literature Review

3.1 University Course Timetabling and Constraints

Babaei, Karimpour and Hadidi (2015) review major approaches to the university course timetabling problem and describe the need to allocate events to time slots and rooms while considering teachers, students, courses and room facilities. The study makes clear that timetable generation is not simply about finding an empty period; several resources have to be available at the same time.

Chen et al. (2021) review research trends, methods and datasets in university course timetabling. They distinguish hard constraints from soft constraints and show the continued use of metaheuristic and hybrid methods. This distinction is useful for SARMS because rules such as avoiding double-booking are treated differently from preferences such as spreading classes across the week.

Bettinelli et al. (2015) provide an overview of curriculum-based course timetabling and discuss exact and heuristic approaches. Bashab et al. (2023) provide a wider review of optimisation techniques, constraints, benchmarks and open issues. Together, these studies support the use of clearly defined constraints and measurable evaluation when developing a timetable system.

3.2 Room Allocation and Real-World Scheduling

Room allocation is closely linked to timetable generation. Phillips et al. (2015) study large practical classroom assignment problems using integer programming and show why room assignment should be treated as a resource-allocation problem rather than as a final manual correction. SARMS follows the same practical idea by checking room capacity and room type while placing a session.

Cruz-Chávez et al. (2016) present a constraint-satisfaction model for a real university case and compare the resulting allocation with an existing institutional schedule. Their work is relevant because it demonstrates how local institutional rules can shape the scheduling model. SARMS takes a similar practical approach, but uses a priority-based constructive heuristic rather than reproducing their method.

3.3 Robustness and Institutional Practice

Oude Vrielink et al. (2019) highlight the importance of connecting timetable research with the way higher-education institutions actually operate. Gülcü and Akkan (2020) study timetable robustness when disruptions affect resources such as teachers or rooms. These studies suggest that timetable

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

quality should not be judged only at the moment of initial generation; a useful system should also be able to respond safely when conditions change.

For SARMS, this supports the planned integration of faculty leave, substitution and other administrative workflows. Those features are treated as development work until they have been tested end to end rather than being presented as complete simply because an interface or database table exists.

4. Research Gap

The literature shows substantial work on university timetabling, optimisation, classroom assignment and timetable robustness (Babaei, Karimpour and Hadidi, 2015; Bettinelli et al., 2015; Chen et al., 2021; Bashab et al., 2023). Several studies use mathematically detailed models or benchmark and institutional datasets (Phillips et al., 2015; Cruz-Chávez et al., 2016; Antunes-Batista et al., 2026). Other work looks at how scheduling is used in higher-education institutions and how timetables can be affected by disruptions (Oude Vrielink et al., 2019; Gülcü and Akkan, 2020).

The gap addressed by SARMS is therefore not that timetable generation has been ignored by research. The practical gap is the need to bring a college's own rules into one manageable system: multiple departments and sections, language groups that regroup students, shared subjects, laboratory requirements, limited rooms and clear diagnostics when the requested timetable cannot be completed.

SARMS addresses this gap through a web application and a transparent scheduling heuristic. Its usefulness should be judged by whether the generated timetable respects required constraints, accounts for required teaching hours, reports unplaced sessions clearly and can be maintained when academic data changes. Claims that SARMS is faster or better than another algorithm would require a controlled comparison and are not made here.

5. Problem Statement

SARMS addresses the problem of assigning required teaching sessions for multiple classes and sections to available days, periods, faculty members and rooms and substitution of the teacher. The system has to consider weekly subject hours, differences between theory and laboratory sessions, language groups, shared subjects, faculty availability, classroom capacity, room type and timetable settings. A valid allocation must avoid simultaneous assignments for the same class, faculty member or room.

Limited resources can make some sessions impossible to place even when the input data are valid. For that reason, the system should not simply display the sessions it managed to place. It should also

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

identify the class, subject, missing hours and likely reason when a session remains unplaced. The problem is therefore twofold: produce a timetable that respects the required constraints and make incomplete or infeasible cases visible so that an administrator can correct the data or reconsider the available resources.

6. Aim and Objectives

Aim: To develop a web-based Smart Academic Resource Management System that supports timetable generation and classroom allocation while reflecting the actual academic scheduling requirements of a college.

- To organise class, subject, faculty, room and language-group information in one central database.
- To generate weekly timetables using a priority-based approach that handles restrictive activities before regular lectures.
- To allocate rooms according to availability, capacity and laboratory requirements.
- To reduce class, faculty and room conflicts through availability checks and independent validation.
- To identify unplaced sessions and provide useful information for corrective action.
- To provide a foundation for faculty assignment, leave, substitution, notifications and other academic-management workflows.
- To define measurable tests for timetable completeness, conflict detection, room suitability and generation time.

7. Research Questions

- RQ1. How can a web-based academic resource management system generate timetables for multiple classes and sections while considering faculty availability and limited classroom resources?
- RQ2. How can language groups, shared subjects and laboratory sessions be scheduled while avoiding conflicts between classes, faculty members and rooms?
- RQ3. How well does the current priority-based heuristic account for room capacity and room suitability during timetable generation?
- RQ4. How can the system identify unplaced sessions and scheduling warnings so that incomplete timetables can be reviewed before use?
- RQ5. How can SARMS be extended to support timetable changes and wider administrative workflows, including faculty leave, substitution and payslip allocation?

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

8. Methodology

8.1 Development Approach

The project follows an applied system-development approach. Requirements were translated into a relational database, a web interface, backend API routes and a timetable-generation module. Instead of trying to implement every optimisation method found in the literature, the project focuses on rules that match the college context. The overall process consists of preparing the data, creating scheduling tasks, placing them according to priority, validating the result and evaluating the output.

8.2 Data Model and Inputs

SARMS uses a MySQL/MariaDB database containing departments, classes and sections, student strength, subjects, weekly subject hours, faculty assignments, language groups, rooms and timetable settings. Room records include capacity and room type, allowing the generator to reject rooms that are too small or unsuitable for a particular session.

8.3 Priority-Based Scheduling Algorithm

The generator uses a constructive heuristic. It places one task at a time rather than searching every possible timetable, so it does not claim global optimality. The current priority order is: (1) language blocks, (2) joint subjects such as shared English sessions, (3) laboratory sessions and (4) regular lectures. Shared and specialised sessions are handled first because they usually have fewer possible placements.

8.5 Evaluation Procedure

Evaluation should use the same database configuration for each test run and record both the generated timetable and the diagnostics returned by the system. Each run should be checked for class overlaps, faculty overlaps, room overlaps, capacity violations, room-type violations, missing weekly hours and correct handling of shared language or joint sessions. The main measures are scheduled-session rate, unplaced hours, detected conflicts, room-capacity or room-type violations and generation time. These are proposed evaluation measures; verified runtime and improvement percentages require logged test runs.

9. System Architecture

SARMS follows a client-server structure. Users interact with the React and Vite frontend, which sends requests to the Node.js and Express.js backend. The backend reads and updates academic records in MySQL/MariaDB and calls the timetable generator when a schedule is requested. The

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

generator reads the required data, places sessions and sends the candidate timetable through validation. The resulting timetable or diagnostic information is returned to the user interface.

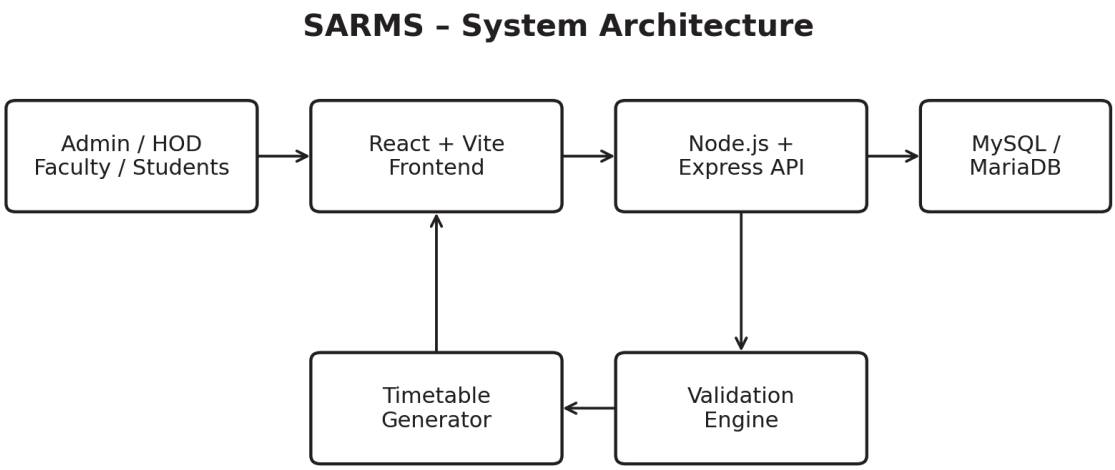


Figure 1. Proposed SARMS architecture

10. Key Findings

The findings currently supported by the prototype are implementation-level findings. They describe what has been built into the scheduling logic and should not be read as measured proof that SARMS is faster or more efficient than manual scheduling.

- The generator uses a priority order that handles language groups first, followed by joint subjects, laboratory sessions and regular lectures.
- Availability checks are built into placement, so occupied classes, occupied faculty and unsuitable or unavailable rooms can be rejected.
- The room-selection logic prefers the smallest available room that can accommodate the group.
- An independent validation stage checks faculty overlaps, room double-booking, class clashes, room capacity and laboratory-room suitability.
- Unplaced sessions can be reported with a reason, making resource shortages and scheduling difficulties visible.
- A timetable can pass conflict checks while still having unplaced required hours, so conflict-free output should not automatically be treated as complete.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

- No numerical claim is made about generation time, percentage of sessions scheduled, reduction in manual work or improvement in room utilisation because those figures require logged and repeatable tests.

11. Discussion

The SARMS design follows a central point found across the reviewed research: a timetable has to coordinate time, people, classes and physical resources together. The priority-based heuristic gives shared language sessions, joint subjects and practical sessions attention before ordinary lectures. The best-fit room rule provides a simple way to handle limited classroom capacity, while the validator checks several non-negotiable constraints (Babaei, Karimpour and Hadidi, 2015; Phillips et al., 2015; Chen et al., 2021).

This approach is useful for a college setting where administrators need rules they can understand and diagnostics they can act on. At the same time, it should not be confused with an exact optimisation model or a multiobjective evolutionary approach. More advanced methods can optimise a wider set of objectives and compare solutions using benchmark or real-world data (Phillips et al., 2015; Antunes-Batista et al., 2026). SARMS currently favours a simpler and more transparent heuristic, which makes it easier to explain but still needs stronger evaluation under tighter resource conditions.

One important distinction is between feasibility and completeness. A timetable may contain no overlaps among the sessions that were placed and still be missing some required teaching hours. For practical use, SARMS should therefore check both the absence of hard-constraint violations and the completion of required hours. This also matters when the timetable has to respond to leave, substitutions or other changes, an issue highlighted in research on robust timetabling (Gülcü and Akkan, 2020).

12. Limitations

- The scheduling method is a constructive heuristic and does not prove global optimality.
- The current validation focuses on placed timetable rows; unplaced hours need a stronger completeness check.
- Fixed break periods are not yet represented as explicit blocked time slots.
- Some shared-subject cases may require clearer rules when different sections have different teachers.
- The current development dataset and testing scale are limited, and there is no controlled comparison with a manual timetable or another algorithm.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

- Faculty leave, substitution, notifications and other administrative workflows require complete end-to-end testing.
- Faculty payslip allocation is not an implemented result; it is proposed as future scope.

13. Future Scope

- To add faculty payslip allocation so verified faculty workload and payment-related records can support a controlled payslip workflow, subject to institutional payroll rules and access permissions.
- To strengthen completeness validation so a timetable with unplaced required hours is clearly marked incomplete.
- To represent fixed breaks, lunch periods, faculty availability windows and other institutional time restrictions explicitly.
- To evaluate stronger optimisation methods, including local search or metaheuristics, and compare them with the current heuristic on the same dataset.
- To extend faculty substitution support by checking subject suitability, faculty availability, workload and approval status before assigning a substitute.
- To test shared language groups and joint subjects across database records and all timetable views.
- To add systematic logging for runtime, scheduled-session rate, unplaced hours, conflicts and room utilisation.
- To conduct a wider pilot across additional semesters and departments and compare SARMS with the institution's existing scheduling process.

14. Conclusion

SARMS is a web-based prototype designed to make college timetable generation and academic resource management more organised and easier to monitor. It combines a React and Vite frontend, a Node.js and Express.js backend and a MySQL/MariaDB database. Its scheduling logic gives priority to language groups and shared subjects before laboratory sessions and regular lectures, checks resource availability and validates several important timetable conflicts.

The main contribution of the project is its practical focus. Instead of treating timetable generation as an isolated algorithm, SARMS brings institution-specific rules, resource limits and diagnostic reporting into one application. This makes the project relevant to digital transformation in education because a process that normally depends heavily on manual coordination can be handled through a central digital system.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

The prototype still needs stronger completeness validation, break handling, end-to-end testing of supporting workflows and measured comparison with the existing manual process. Future work will also cover substitution support, stronger optimisation and faculty payslip allocation. With these additions and a larger, repeatable evaluation, SARMS can move from a working prototype toward a more dependable academic resource-management platform.

References

Antunes-Batista, B., Atta, S., Basto-Fernandes, V., Yevseyeva, I. and Emmerich, M. (2026) 'Multiobjective optimization approaches for room allocation in university course timetabling', Procedia Computer Science, 278, pp. 663–670. Available at: https://doi.org/10.1016/j.procs.2026.03.037.

Babaei, H., Karimpour, J. and Hadidi, A. (2015) 'A survey of approaches for university course timetabling problem', Computers & Industrial Engineering, 86, pp. 43–59. Available at: https://doi.org/10.1016/j.cie.2014.11.010.

Bashab, A., Ibrahim, A.O., Tarigo Hashem, I.A., Aggarwal, K., Mukhlif, F., Ghaleb, F.A. and Abdelmaboud, A. (2023) 'Optimization techniques in university timetabling problem: constraints, methodologies, benchmarks, and open issues', Computers, Materials & Continua, 74(3), pp. 6461–6484. Available at: https://doi.org/10.32604/cmc.2023.034051.

Bettinelli, A., Cacchiani, V., Roberti, R. and Toth, P. (2015) 'An overview of curriculum-based course timetabling', TOP, 23(2), pp. 313–349. Available at: https://doi.org/10.1007/s11750-015-0366-z.

Chen, M.C., Sze, S.N., Goh, S.L., Sabar, N.R. and Kendall, G. (2021) 'A survey of university course timetabling problem: perspectives, trends and opportunities', IEEE Access, 9, pp. 106515–106529. Available at: https://doi.org/10.1109/ACCESS.2021.3100613.

Cruz-Chávez, M.A., Flores-Pichardo, M., Martínez-Oropeza, A., Moreno-Bernal, P. and Cruz-Rosales, M.H. (2016) 'Solving a real constraint satisfaction model for the university course timetabling problem: a case study', Mathematical Problems in Engineering, 2016, Article ID 7194864. Available at: https://doi.org/10.1155/2016/7194864.

Gülcü, A. and Akkan, C. (2020) 'Robust university course timetabling problem subject to single and multiple disruptions', European Journal of Operational Research, 283(2), pp. 630–646. Available at: https://doi.org/10.1016/j.ejor.2019.11.024.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

Oude Vrielink, R.A., Jansen, E.A., Hans, E.W. and van Hillegersberg, J. (2019) 'Practices in timetabling in higher education institutions: a systematic review', Annals of Operations Research, 275(1), pp. 145–160. Available at: https://doi.org/10.1007/s10479-017-2688-8.

Phillips, A.E., Waterer, H., Ehrgott, M. and Ryan, D.M. (2015) 'Integer programming methods for large-scale practical classroom assignment problems', Computers & Operations Research, 53, pp. 42–53. Available at: https://doi.org/10.1016/j.cor.2014.07.012.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)