International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Smart Classroom and Timetable Scheduler

D. Renuka Asst.
Professor
Department of Computer Science and Engineering
Mahatma Gandhi Institute of Technology
Hyderabad 500075, India
drenuka_cse@mgit.ac.in

Dr. B. Madhava
Rao Asst.
Professor
Department of Computer Science and Engineering
Mahatma Gandhi Institute of Technology
Hyderabad 500075, India
bommineni6170@gmail.com

R. Ajay
Student
Department of Computer Science and Engineering
Mahatma Gandhi Institute of Technology
Hyderabad 500075, India
rajay_cse2405x4@mgit.ac.in

R.Enosh Anshul
Benjamin Student
Department of Computer Science and Engineering
Mahatma Gandhi Institute of Technology
Hyderabad 500075, India
renoshanshulbenjamin_cse2405x3@mgit.ac.in

***Abstract*—This paper presents a Smart Classroom and Timetable Scheduler designed to automate and optimize academic scheduling in educational institutions. Traditional timetable generation is a complex and time-consuming process that often leads to conflicts such as faculty overlaps, improper classroom allocation, and uneven subject distribution. The proposed system addresses these challenges by integrating constraint-based scheduling with a round-robin algorithm to ensure efficient and conflict-free timetable generation. The system considers multiple constraints, including faculty availability, classroom capacity, subject hours per week, and lab session continuity. Additionally, it provides a user-friendly web interface that allows administrators to manage academic data, generate schedules, and export timetables in various formats. The implementation demonstrates improved scheduling accuracy, reduced manual effort, and enhanced resource utilization. This system contributes to the development of intelligent academic management solutions by combining algorithmic efficiency with practical usability.**

***Index Terms*—Timetable Scheduling, Smart Classroom, Constraint-Based Scheduling, Round Robin Algorithm, Academic Automation, Resource Optimization.**

I. INTRODUCTION

THE Timetable scheduling is a fundamental and complex task in educational institutions, involving the systematic allocation of subjects, faculty members, classrooms, and time slots in a structured manner. The primary The Smart Classroom and Timetable Scheduler proposed in this work is designed to automate the entire scheduling objective of timetable generation is to ensure that all academic activities are conducted efficiently without any conflicts or resource wastage. However, manual timetable preparation is a challenging and time-consuming process that often leads to issues such as faculty overlaps, classroom conflicts, uneven subject distribution, and inefficient utilization of available resources. As the number of courses, departments, and faculty members increases, the complexity of scheduling grows significantly.

Managing multiple constraints such as faculty availability, classroom capacity, subject hour requirements, and laboratory sessions becomes increasingly difficult using traditional approaches. In many institutions, timetable generation still relies on manual effort or semi-automated tools, which are prone to errors and lack scalability. To address these challenges, automated timetable scheduling systems have gained importance in recent years. These systems aim to reduce human intervention, improve accuracy, and optimize resource allocation. Various computational approaches such as constraint-based scheduling, optimization algorithms, and artificial intelligence techniques have been explored to solve the timetabling problem. Among these, constraint-based methods are widely used due to their ability to strictly enforce scheduling rules, while heuristic techniques like round-robin help in achieving balanced distribution.

process using a combination of constraint-based validation and round-robin allocation. The system ensures that all hard

This work was carried out as part of the Bachelor of Technology (B.Tech) program in the Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology (MGIT), Hyderabad, India.

R. Enosh Anshul Benjamin is with the Department of Computer Science and Engineering, MGIT, Hyderabad, India (e-mail: renoshanshulbenjamin_cse2405x3@mgit.ac.in).

R. Ajay is with the Department of Computer Science and Engineering, MGIT, Hyderabad, India (e-mail: rajay_cse2405x4@mgit.ac.in).

This work has no external funding or sponsorship.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

constraints, such as no faculty clash and no classroom overlap. It provides a scalable and reliable solution that can be effectively implemented in educational institutions to improve scheduling accuracy, reduce manual workload, and ensure optimal utilization of resources.

II. GUIDELINES FOR SYSTEM DESIGN AND IMPLEMENTATION

The design and implementation of the Smart Classroom and Timetable Scheduler focus on ensuring accuracy, efficiency, and usability in academic scheduling. The system is developed using a structured approach that combines constraint-based scheduling with a round-robin technique to generate optimized timetables. The system ensures proper handling of academic data such as subjects, faculty, classrooms, and lab sessions. It strictly follows constraints like no faculty overlap, no classroom conflict, and fixed subject hours per week. These constraints are validated before assigning any subject to a time slot, ensuring a conflict-free timetable.

The implementation also emphasizes clarity and simplicity in user interaction. A web-based interface is provided for administrators to manage academic data and generate timetables easily. Faculty and students can view their schedules in a structured and user-friendly format. Special attention is given to handling laboratory sessions, which require continuous time blocks. The system ensures that lab sessions are allocated without interruption, maintaining practical feasibility.

Overall, the system design aims to provide a reliable, scalable, and efficient solution for academic timetable generation while minimizing manual effort and errors.

III. LITERATURE SURVEY

Literature survey plays a crucial role in understanding the existing methodologies, technologies, and challenges involved in timetable scheduling systems. It provides insights into different approaches used for automating scheduling tasks, including constraint-based techniques, optimization methods, and intelligent algorithms. The survey helps in identifying limitations of existing systems and guides the development of an efficient, scalable, and user-friendly solution.

R. K. Thakur et al., [1] presented a practical approach to college timetable scheduling using the Constraint Satisfaction Problem (CSP) technique. The study highlights how constraint-based methods can effectively automate timetable generation by handling conditions such as teacher availability, classroom allocation, and subject distribution. The system ensures that no conflicts occur between faculty members, classrooms, and subjects by strictly validating constraints before scheduling. This approach significantly reduces manual effort and improves scheduling accuracy. However, it requires precise constraint data and becomes difficult to scale for large institutions with complex scheduling requirements.

Xin Gu, [2] reviewed modern techniques for solving university timetabling problems, focusing on methods such as Integer Programming and Machine Learning. The study explains how these techniques optimize resource allocation and improve scheduling efficiency by minimizing conflicts. Machine learning models can adapt to dynamic scheduling scenarios and provide intelligent solutions based on historical data. Despite their advantages, these methods involve complex mathematical modeling and require high computational resources, making them less practical for smaller or real-time systems.

M. Davison et al., [3] focused on solving the university course timetabling problem by incorporating hybrid teaching considerations, including both online and offline classes. The proposed approach uses constraint-based optimization to ensure that hard constraints such as no class overlap, room capacity limits, and faculty availability are satisfied. Additionally, soft constraints are used to improve timetable quality and flexibility. While this method produces efficient and adaptable schedules, it requires advanced modeling techniques and significant computational effort, which may limit its implementation in simple systems.

In addition to these approaches, heuristic techniques such as the round-robin method have been widely used for distributing tasks evenly across available resources. These methods are simple to implement and ensure balanced allocation of subjects across time slots. However, they do not inherently consider constraints and must be combined with validation techniques to avoid scheduling conflicts.

TABLE I
LITERATURE SURVEY

Author & Year	**Method Used**	**Advantages**	**Limitations**
Thakur (2024)	Constraint Satisfaction Problem (CSP)	Ensures conflict-free scheduling	Difficult to scale
Xin Gu (2025)	ML + IP	Improves scheduling efficiency	Complex
Davison (2025)	Optimization	Flexible scheduling	High cost

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

IV. PROPOSED SYSTEM

The proposed Smart Classroom and Timetable Scheduler is a web-based application designed to automate and optimize the process of academic timetable generation. The system aims to eliminate the challenges associated with manual scheduling, such as faculty conflicts, classroom overlaps, and inefficient distribution of subjects across time slots. By integrating intelligent scheduling techniques, the system ensures accuracy, efficiency, and scalability.

The system accepts essential academic inputs, including subject details, faculty information, classroom data, and laboratory session requirements. These inputs are processed using a hybrid scheduling approach that combines constraint-based validation with a round-robin distribution technique. The constraint-based method ensures that all hard constraints —such as no teacher overlap, no classroom conflict, and fixed subject hours—are strictly followed. At the same time, the round-robin technique ensures balanced distribution of subjects across the timetable, avoiding clustering and repetition.

A key feature of the system is its ability to handle laboratory sessions effectively. Unlike regular classes, lab sessions require continuous time blocks. The system prioritizes lab allocation by pre-assigning them into consecutive time slots before scheduling theory subjects. This ensures that practical sessions are conducted without interruption and align with real-world academic requirements.

The system follows a structured scheduling flow. Initially, all lab sessions are placed in the timetable. Then, a subject pool is created based on the number of hours required per week for each subject. Using the round-robin approach, subjects are assigned sequentially to available slots while continuously checking constraints. If a conflict is detected, the system automatically selects an alternative subject from the pool, ensuring that scheduling continues without interruption.

Another important aspect of the proposed system is its user-friendly interface. The system provides separate dashboards for administrators, faculty, and students. Administrators can input and manage academic data, generate timetables, and export schedules in formats such as PDF and Excel. Faculty members can view their teaching schedules, while students can access their class timetables in a structured and easy-to-understand format.

The system is designed to be dynamic and flexible. It allows regeneration of timetables whenever changes occur, such as faculty updates or classroom modifications. Additionally, it ensures efficient utilization of available resources by minimizing idle time slots and evenly distributing workload among faculty members.

Compared to existing systems, the proposed approach offers a balance between simplicity and efficiency. While advanced optimization techniques may provide highly optimized solutions, they often involve complex implementation and high computational cost. The proposed system achieves reliable performance using a simpler yet effective combination of constraint-based and heuristic techniques.

Overall, the Smart Classroom and Timetable Scheduler provides a practical, scalable, and efficient solution for academic scheduling. It reduces manual effort, improves accuracy, and enhances the overall management of academic resources, making it highly suitable for modern educational institutions. sessions are pre-placed into the timetable to avoid fragmentation and ensure proper allocation of practical classes.

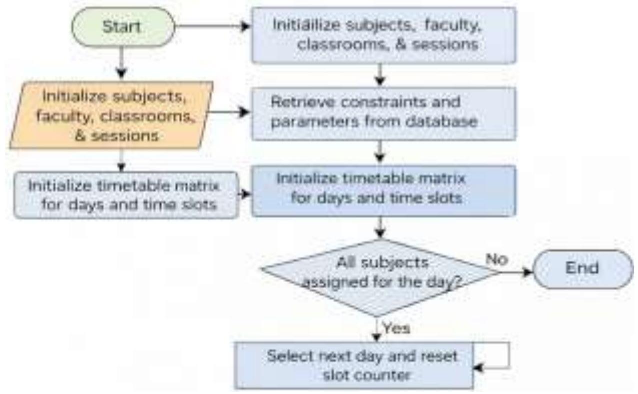

Fig. 1. Timetable Generation Flowchart

DESIGN AND METHODOLOGY

The methodology of the proposed Smart Classroom and Timetable Scheduler is designed to systematically automate the process of timetable generation using a combination of structured data processing and intelligent scheduling techniques. The system follows a step-by-step workflow that begins with data acquisition and ends with the generation of an optimized, conflict-free timetable. Each stage of the methodology plays a crucial role in ensuring accuracy, efficiency, and reliability of the system.

A. Constraint-Based Scheduling

The first stage involves data collection, where the system gathers essential academic information such as subjects, faculty details, classroom availability, and laboratory requirements. This data is provided through a web-based interface, ensuring ease of access and usability. The system stores this information in a structured database format, enabling efficient retrieval and processing during timetable generation.

B. Round Robin Technique

Once the data is collected, the preprocessing stage is carried out to validate and organize the input. This includes checking for missing values, verifying subject hours per week, ensuring faculty assignments are correct, and confirming classroom capacities. Data validation is an important step as it prevents errors during scheduling and ensures that all constraints are logically consistent before proceeding further. The core of the methodology lies in the scheduling algorithm, which integrates constraint-based scheduling with a round-robin distribution approach.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Initially, the system identifies and allocates laboratory sessions since they require continuous time blocks. These
Data Validation Technique

After lab allocation, the system generates a subject pool based on the required number of hours for each subject. The round-robin technique is then applied to distribute subjects evenly across available time slots. This ensures that no subject is repeatedly assigned in consecutive slots and maintains a balanced schedule throughout the week. At each step, the system checks for constraints such as no faculty overlap,no classroom duplication, and valid time slot assignment. If a conflict is detected during scheduling, the system dynamically adjusts by selecting an alternative subject from the pool. This adaptive behavior ensures continuity in timetable generation without violating constraints. The process continues iteratively until all subjects are assigned or no valid slots remain. Any unassigned subjects are flagged for review, indicating limitations such as insufficient time slots.

The final stage of the methodology involves timetable generation and visualization. The generated schedule is displayed in a structured grid format, making it easy for users to interpret. The system also supports exporting timetables in formats like PDF and Excel, enabling easy sharing and documentation. Additionally, different user roles such as administrators, faculty, and students are provided with customized views to access relevant information efficiently.

Overall, the proposed methodology ensures a well-organized, automated, and scalable approach to timetable generation. By combining constraint validation with heuristic scheduling techniques, the system achieves a balance between performance and simplicity. This structured methodology significantly reduces manual effort, minimizes scheduling conflicts, and enhances the effective utilization of academic resources.

C. Complexity Analysis

The algorithm iterates through all available time slots across all days, resulting in a time complexity of $O(D \times S)$. For each slot, the system selects a suitable subject from the subject pool while ensuring constraint validation. In the worst case, this involves scanning the subject pool, leading to an additional factor of $O(N)$.

Thus, the overall time complexity of the scheduling process can be approximated as $O(D \times S \times N)$. However, since the number of subjects is relatively limited in practical scenarios, the algorithm performs efficiently in real-time applications.

The space complexity is $O(N)$, as the system maintains a subject pool and a schedule matrix to store assigned timetable entries.

Overall, the algorithm is computationally efficient and suitable for real-time timetable generation in educational institutions.

V. SYSTEM IMPLEMENTATION

The Smart Classroom and Timetable Scheduler is implemented as a web-based application using modern technologies to ensure efficiency, scalability, and ease of use. The system follows a modular architecture consisting of frontend, backend, and database components that work together to automate timetable generation. The frontend of the system is developed using React.js, which provides a responsive and interactive user interface. It allows users such as administrators, faculty, and students to interact with the system through different dashboards. The interface includes forms for adding subjects, classrooms, and lab sessions, along with visualization components for displaying generated timetables in a structured grid format. The use of modern UI libraries and animations enhances user experience and makes the system visually intuitive.

The backend is implemented using Node.js and Express.js, which handle the core logic of the application. The backend is responsible for processing user inputs, managing API requests, and executing the timetable generation algorithm. It ensures proper communication between the frontend and the database. RESTful APIs are used to perform operations such as adding data, retrieving records, generating schedules, and exporting timetables. The database used in the system is MongoDB, which stores all academic data including subjects, faculty details, classrooms, lab sessions, and generated timetables. The data is structured in collections, allowing efficient storage and retrieval. MongoDB’s flexibility helps in handling dynamic data and makes it suitable for scalable applications.

The core functionality of the system lies in the timetable generation module. The implementation follows a hybrid approach where lab sessions are first allocated into continuous time slots. After this, subjects are arranged into a pool based on their required weekly hours. A round-robin algorithm is then applied to assign subjects sequentially to available time slots. During this process, constraint validation is continuously performed to ensure there are no conflicts such as overlapping classes, duplicate faculty assignments, or invalid room allocations. The system also includes data validation mechanisms to ensure the correctness of input data. Before generating the timetable, the system checks whether all required fields are filled and whether constraints such as subject hours and faculty assignments are valid. This reduces errors and improves the reliability of the generated schedule.

Additionally, the system provides export functionality, allowing users to download timetables in PDF and Excel formats. This feature is implemented in the backend and store

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

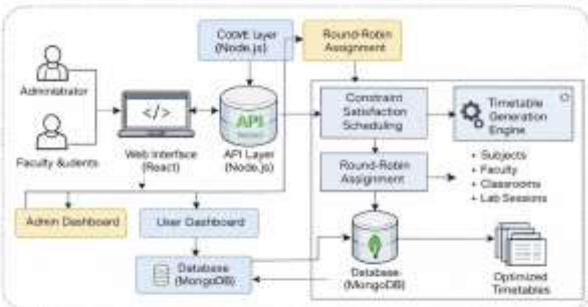

Fig. 2. System Architecture

schedules. Security is handled through authentication and authorization mechanisms. Users are required to log in to access the system, and role-based access control ensures that only authorized users can perform specific actions such as generating or deleting timetables.

Overall, the implementation of the system integrates modern web technologies with efficient scheduling logic to provide a reliable and user-friendly solution. The modular design ensures that the system can be easily extended in the future, while the use of automated algorithms significantly reduces manual effort and improves scheduling accuracy.

VI. RESULTS AND DISCUSSION

The Smart Classroom and Timetable Scheduler was implemented and tested under various input conditions to evaluate its performance, accuracy, and reliability. The system was provided with different sets of data, including multiple subjects, faculty members, classrooms, and laboratory sessions, to simulate real-world academic scheduling scenarios. The results demonstrate that the system successfully generates structured and conflict-free timetables while satisfying all predefined constraints. One of the key outcomes observed is the elimination of common scheduling conflicts such as faculty overlap and classroom clashes. The constraint-based validation mechanism ensures that no teacher is assigned to more than one class at the same time and that no classroom is allocated to multiple subjects simultaneously. This significantly improves the reliability of the timetable compared to manual scheduling methods. The system also ensures balanced distribution of subjects across the week using the round-robin technique. Subjects are evenly allocated to available time slots, avoiding repetition in consecutive periods and preventing overloading of specific days. This results in a well-organized timetable that maintains academic balance and improves learning efficiency. Laboratory sessions are handled effectively by allocating continuous time slots. The system prioritizes lab scheduling and ensures that practical sessions are conducted without interruption. This feature reflects real-world academic requirements and enhances the usability of the system.

The performance of the system was evaluated based on execution time and efficiency. The timetable generation process is completed within a short duration, even with multiple subjects and constraints. Compared to manual scheduling, the system significantly reduces time and effort while improving accuracy. The automated approach minimizes human errors and ensures consistency in scheduling.

The user interface also plays a crucial role in the overall effectiveness of the system. The generated timetable is displayed in a clear and structured format, making it easy for users to understand and interpret. Features such as PDF and Excel export further enhance usability by allowing users to store and share timetables conveniently. Additionally, the system supports dynamic updates, enabling users to regenerate timetables whenever changes occur. This flexibility makes the system adaptable to real-time requirements such as faculty changes or classroom availability updates. The ability to handle such changes efficiently demonstrates the robustness of the proposed solution.

Overall, the results indicate that the Smart Classroom and Timetable Scheduler is an efficient, reliable, and scalable solution for academic scheduling. It improves resource utilization, reduces manual workload, and provides a practical approach for managing complex scheduling tasks in educational institutions.

VII. CONCLUSION AND FUTURE SCOPE

A. Conclusion

The Smart Classroom and Timetable Scheduler successfully demonstrates an efficient and automated approach to academic timetable generation. The system addresses the major challenges associated with manual scheduling, such as faculty clashes, classroom conflicts, uneven subject distribution, and excessive time consumption. By integrating constraint-based scheduling with a round-robin allocation technique, the system ensures that all predefined rules are strictly followed while maintaining a balanced and optimized timetable structure. The implementation of the system highlights its ability to generate conflict-free schedules by validating constraints at every stage of the scheduling process. It ensures that no faculty member is assigned multiple classes simultaneously, no classroom is double-booked, and all subjects are allocated their required number of hours per week. Additionally, the proper handling of laboratory sessions through continuous slot allocation reflects real-world academic requirements and enhances the practicality of the system. The user-friendly interface further improves the effectiveness of the system by allowing administrators to input data, generate timetables, and manage academic resources easily. Faculty members and students can access their schedules in a structured format, improving communication and academic planning. The inclusion of export features such as PDF and Excel adds convenience and usability.

Overall, the proposed system provides a reliable, scalable, and efficient solution for academic scheduling. It significantly reduces manual effort, minimizes human errors, and improves resource utilization. The combination of simplicity and effectiveness makes the system suitable for

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

implementation in educational institutions of varying sizes.

B. Future Scope

The Smart Classroom and Timetable Scheduler can be further enhanced by incorporating advanced technologies and features to improve its functionality and adaptability. One of the major future improvements is the integration of Artificial Intelligence and Machine Learning techniques to develop predictive and adaptive scheduling systems. These techniques can analyze historical data and user preferences to generate more optimized and intelligent timetables.

Another important enhancement is the implementation of real-time scheduling capabilities. This would allow the system to dynamically update timetables in response to sudden changes such as faculty unavailability, classroom maintenance, or schedule adjustments. Real-time updates would make the system more flexible and suitable for practical use in dynamic academic environments. The system can also be extended by developing mobile applications for both Android and iOS platforms. This would enable students and faculty members to access their schedules anytime and receive instant notifications about changes, improving accessibility and communication. In addition, advanced optimization techniques such as genetic algorithms, swarm intelligence, or heuristic search methods can be incorporated to further improve scheduling efficiency, especially for large-scale institutions with complex constraints. These techniques can help in generating near-optimal timetables with better resource utilization. Cloud-based deployment is another potential enhancement, allowing the system to support multiple institutions simultaneously. This would improve scalability, data accessibility, and system reliability. Integration with institutional management systems such as ERP platforms can further streamline academic operations.

Finally, the system can include analytical and reporting features to monitor classroom utilization, faculty workload, and scheduling efficiency. These insights can help administrators make better decisions and improve overall academic planning.

VIII. LIMITATIONS

The proposed Smart Classroom and Timetable Scheduler demonstrates effective automation of academic scheduling; however, several limitations are identified that may affect its performance in highly dynamic or large-scale environments. The system primarily operates on predefined constraints and heuristic logic, which limits its ability to adapt automatically to changing academic conditions such as sudden faculty unavailability or real-time classroom adjustments.

Additionally, the current scheduling approach does not incorporate advanced optimization or learning-based techniques, such as genetic algorithms, reinforcement learning, or predictive analytics, which could enhance solution optimality and adaptability. As a result, while the system ensures conflict-free scheduling, it may not always generate globally optimal timetables in terms of resource utilization or workload balancing.

Another limitation is related to scalability. As the number of subjects, departments, and constraints increases, the computational complexity may lead to increased execution time, potentially affecting system responsiveness in large institutions. Furthermore, the system assumes accurate and complete input data; any inconsistencies or missing information may impact the quality of the generated timetable.

The system also provides limited support for multi-department or inter-department scheduling scenarios, where shared resources and cross-department constraints become significant. Moreover, real-time rescheduling capabilities are minimal, restricting flexibility in handling last-minute changes.Lastly, the current implementation is designed as a standalone academic scheduling tool and does not fully support integration with other institutional systems such as attendance tracking, examination scheduling, or student performance analytics. Addressing these limitations would significantly enhance the robustness, intelligence, and scalability of the system.

IX. APPLICATIONS

The Smart Classroom and Timetable Scheduler has broad applicability across various academic and organizational domains, offering significant improvements in efficiency, accuracy, and resource management. In educational institutions such as schools, colleges, and universities, the system can automate the complex process of timetable generation, eliminating manual errors and ensuring optimal allocation of faculty, classrooms, and time slots. It is particularly beneficial in institutions with large-scale academic structures, where traditional scheduling methods are time-consuming and prone to conflicts.

Beyond academic institutions, the system can be effectively applied in coaching centers, training institutes, and professional certification organizations, where scheduling of classes, sessions, and workshops is required on a regular basis. Its ability to manage constraints and distribute sessions evenly makes it highly suitable for environments that require structured planning and efficient time utilization.

In corporate environments, the system can be adapted for managing employee training schedules, meeting room allocations, and workshop planning. Organizations conducting frequent training programs or seminars can leverage the system to streamline scheduling operations and improve resource utilization. The flexibility of the system allows it to be customized according to organizational requirements.

The system also has strong potential for integration with institutional Enterprise Resource Planning (ERP) systems, enabling seamless coordination between various academic processes such as student enrollment, faculty workload management, classroom utilization, and attendance tracking.

Such integration can lead to the development of a unified academic management platform, improving overall operational efficiency.

With further enhancements, the system can support real-time scheduling and dynamic updates, allowing it to adapt instantly to changes in academic environments. The addition of mobile applications can provide accessibility to students and faculty, enabling them to view schedules, receive notifications, and stay updated with minimal effort.

Moreover, incorporating analytical and reporting modules can enable administrators to monitor key metrics such as classroom utilization, faculty workload distribution, and scheduling efficiency. These insights can support data-driven decision-making and long-term academic planning.

Overall, the Smart Classroom and Timetable Scheduler is a versatile and scalable solution that can be extended beyond traditional educational systems to various scheduling-intensive domains, making it a valuable tool in modern digital.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology (MGIT), Hyderabad, for providing the necessary facilities and support to carry out this project successfully. We would like to extend our heartfelt thanks to our project guide for their valuable guidance, continuous encouragement, and constructive suggestions throughout the development of this work. Their support played a crucial role in the successful completion of this project. We also express our appreciation to all the faculty members and friends who provided assistance and motivation during the course of this project. Finally, we are grateful to our family members for their constant support and encouragement.

D. Renuka is an Assistant Professor in the Department of Computer Science and Engineering at Mahatma Gandhi Institute of Technology (MGIT), Hyderabad, India. She holds a Master of Technology (M.Tech) degree in Computer Science and Engineering. She has over 7 years of teaching experience and has handled subjects such as Data Structures, Operating Systems, Database Management Systems, Web Technologies, and Design and Analysis of Algorithms. Her research interests include machine learning, deep learning, and optimization techniques. She has published research work in reputed journals and conferences and is a life member of professional bodies such as ISTE and IAENG. She has received several recognitions, including UGC-NET qualification and NPTEL Elite certification. She is actively involved in mentoring students and guiding academic projects.

Dr. B. Madhava Rao, is an Assistant Professor in the Department of Computer Science and Engineering at Mahatma Gandhi Institute of Technology (MGIT), Hyderabad, India. He holds a Ph.D. in Computer Science and Engineering with specialization in Quantum Computing. He has over 18 years of academic and industry experience and has taught a wide range of undergraduate and postgraduate subjects including Machine Learning, Operating Systems, Computer Networks, and Data Analytics. His research interests include quantum computing, cybersecurity, and intelligent systems. Dr. Rao has published multiple research papers, authored books, and holds patents in emerging technology domains. He is also a life member of ISTE and actively contributes to academic research and development activities.

R. Enosh Anshul Benjamin, is currently pursuing a Bachelor of Technology degree in Computer Science and Engineering at Mahatma Gandhi Institute of Technology (MGIT), Hyderabad, India. His major field of study includes data structures, software engineering, and full-stack development. He has worked on several academic projects involving system design and application development. His research interests include web technologies, database management systems, and software optimization techniques. He contributed to the implementation and backend development of the Smart Classroom and Timetable Scheduler system.

R. Ajay, is currently pursuing a Bachelor of Technology degree in Computer Science and Engineering at Mahatma Gandhi Institute of Technology (MGIT), Hyderabad, India. His major field of study includes software development, web technologies, and intelligent systems. He is actively involved in developing projects related to academic automation and smart scheduling systems. His areas of interest include artificial intelligence, full-stack web development, and optimization algorithms for solving real-world problems. He has contributed to the design and development of the Smart Classroom and Timetable Scheduler system, focusing on efficient timetable generation and user-friendly interface design. He is interested in building innovative and scalable software solutions and aims to pursue advanced research in intelligent systems and automation technologies.

REFERENCES

1. *[1] R. K. Thakur, N. K. Agrawal, and P. Kumar, "A Practical Approach to College Timetable Scheduling," Mathematical Modeling and Computing Journal, vol. 11, no. 2, pp. 120–128, Apr. 2024.*
2. *[2] X. Gu, "From Integer Programming to Machine Learning: A Technical Review on Solving University Timetabling Problems," Computers, vol. 14, no. 1, Jan. 2025, Art. no. 101, doi: 10.3390/ computers14010101.*
3. *[3] M. Davison, J. Smith, and L. Brown, "Modelling and Solving the University Course Timetabling Problem with Hybrid Teaching Considerations," IEEE Access, vol. 12, pp. 34567–34578, Mar. 2026.*
4. *[4] E. K. Burke and S. Petrovic, "Recent research directions in automated timetabling," European Journal of Operational Research, vol. 140, no. 2, pp. 266–280, 2002.*
5. *[5] P. Baptiste, C. Le Pape, and W. Nuijten, Constraint-Based Scheduling: Models and Algorithms. New York, NY, USA: Springer, 2001.*
6. *[6] S. Abdullah, E. K. Burke, and B. McCollum, "A hybrid evolutionary approach to the university course timetabling problem," in Proc. IEEE Congr. Evol. Comput., Barcelona, Spain, 2010, pp. 1–8.*

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

7. *[7] A. Schaerf, "A survey of automated timetabling," Artificial Intelligence Review, vol. 13, no. 2, pp. 87–127, 1999.*

8. *[8] R. Lewis, "A survey of metaheuristic-based techniques for university timetabling problems," OR Spectrum, vol. 30, no. 1, pp. 167–190, 2008.*

9. *[9] "React Documentation," Meta Platforms, Inc., 2024. [Online]. Available: https://react.dev*

10. *[10] "Node.js Documentation," OpenJS Foundation, 2024. [Online]. Available: https://nodejs.org*

11. *[11] "MongoDB Documentation," MongoDB Inc., 2024. [Online]. Available: https://www.mongodb.com*

12. *[12] "Express.js Guide," OpenJS Foundation, 2024. [Online]. Available: https://expressjs.com*

13. *[13] "Timetable Scheduling System," GeeksforGeeks, 2024. [Online]. Available: https://www.geeksforgeeks.org*

14. *[14] J. Holland, Adaptation in Natural and Artificial Systems. Ann Arbor, MI, USA: Univ. Michigan Press, 1975.*

15. *[15] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Boston, MA, USA: Addison-Wesley, 1989.* U.S. Department of Health and Human Services, Substance Abuse and Mental Health Services Administration, Office of Applied Studies, doi: 10.3886/ICPSR30122.v2.