

Smart Timetable Generator: Using Rule-Based Constraint Scheduling

Soumya Das*, Swati Kesarwani**, Trupti Hote***, Sujata Kullur****

*(Department of Information Technology, Usha Mittal Institute of Technology, SNDT Women's University, Mumbai, das.soumya0909@gmail.com)

** (Department of Information Technology, Usha Mittal Institute of Technology, SNDT Women's University, Mumbai, kesarwaniswati123@gmail.com)

*** (Department of Information Technology, Usha Mittal Institute of Technology, SNDT Women's University, Mumbai, truptihote07@gmail.com)

**** (Usha Mittal Institute of Technology, SNDT Women's University, Mumbai, sujata.kullur@umit.sndt.ac.in)

Abstract:

Academic timetabling is a complex combinatorial optimization problem that involves allocating instructors, class-rooms, laboratories, and time slots while satisfying numerous institutional constraints [3]. In multi-department academic environments, shared faculty resources, laboratory continuity requirements, and batch-wise divisions significantly increase scheduling complexity, making manual timetable preparation inefficient, time-consuming, and prone to conflicts [2]. This paper presents a Smart Timetable Generator that employs a deterministic rule-based scheduling strategy [4] to systematically enforce hard constraints such as prevention of faculty and room overlaps, maintenance of consecutive laboratory sessions, and adherence to institutional timing structures [5]. In addition, soft constraints including balanced weekly lecture distribution and minimization of idle periods are incorporated to enhance overall timetable quality [6]. The system prioritizes structured allocation, supports parallel resource utilization where feasible, and ensures global consistency when shared resources span multiple branches. By integrating persistent database management, real-time timetable visualization, and controlled rescheduling capabilities [7], the proposed framework provides a scalable, transparent, and practically deployable solution for reliable academic schedule generation in structured higher education institutions.

Keywords — Timetable Generation, Constraint-Based Scheduling, Resource Allocation, Optimization

I. INTRODUCTION

Timetable generation in academic institutions is a complex and time-consuming process due to the presence of multiple hard and soft constraints that must be satisfied simultaneously [7]. In structured higher education environments involving multiple departments, faculty members, classrooms, and laboratories, manual timetable preparation becomes increasingly challenging. The scheduling process must carefully balance faculty availability, room allocation, subject hour requirements, and student

schedules to prevent conflicts and resource underutilization [3], [10]. From a computational perspective, timetable generation is recognized as an NP-hard problem, meaning that optimizing large-scale instances using traditional manual methods is inefficient and often impractical [7].

Hard constraints such as preventing faculty or room clashes and ensuring required subject hours must strictly be satisfied, while soft constraints including balanced lecture distribution and minimization of idle time contribute to overall schedule quality [1], [12]. Manual approaches frequently struggle to

handle these interdependent constraints effectively, leading to inefficiencies and repeated revisions [2], [4]. Therefore, there is a need for an automated, constraint-aware scheduling system to ensure conflict-free, optimized, and practical timetable generation.

A. OBJECTIVE

This paper proposes a Smart Timetable Generator based on a constraint-driven scheduling framework. The system automatically generates conflict-free academic timetables while ensuring optimal allocation of faculty, classrooms, and time slots. It effectively satisfies strict institutional constraints and improves overall scheduling efficiency.

B. CONTRIBUTION

This work proposes an innovative web-based Smart Timetable Generator that introduces a structured and constraint-aware scheduling mechanism for any academic institution. The system automates the complex timetable generation process by efficiently resolving conflicts, optimizing faculty and room allocation, and ensuring balanced academic distribution. By integrating dynamic adjustments and interactive visualization within a unified platform, the proposed solution delivers an efficient, scalable, and practically deployable scheduling framework.

II. PROBLEM STATEMENT

Academic timetable generation in higher education institutions is a complex combinatorial scheduling problem involving multiple interdependent constraints such as faculty availability, classroom allocation, laboratory resources, batch divisions, and institutional policies like fixed lunch breaks. The university course timetabling problem is widely recognized as an NP-hard optimization problem due to its combinatorial explosion of feasible solutions and constraint dependencies [7].

Traditional manual timetable preparation is time-consuming, error-prone, and difficult to modify once generated. Even small changes, such as faculty unavailability, room reassignment, or batch reshuffling, can propagate conflicts across multiple

schedules. Such cascading conflicts have been highlighted in prior studies on automated and large-scale timetabling systems [3], [10]. This often results in teacher clashes (a teacher assigned to two classes at the same time), improper lab allocation, non-consecutive lab periods, or violations of institutional timing constraints.

Laboratory scheduling introduces additional complexity. Unlike lecture sessions that occupy a single period, laboratory sessions typically require consecutive multi-hour blocks and are often conducted in parallel batches under shared supervision. Research in high school and university timetabling has emphasized the difficulty of handling such multi-period and batch-based constraints efficiently [1], [8]. These lab blocks must:

- Occupy strictly consecutive time slots
- Avoid crossing predefined institutional boundaries such as lunch breaks
- Ensure no teacher or lab room is double-booked
- Support parallel execution when multiple batches share the same lab infrastructure

Moreover, in multi-branch environments, ensuring global consistency across all branch timetables is critical. A scheduling decision in one branch must not introduce conflicts in another branch that shares the same faculty or infrastructure resources. Studies on constraint-based and hybrid timetabling approaches have shown that global resource synchronization significantly increases computational complexity. [3], [7].

Existing timetable systems often either:

- Focus purely on automated generation using heuristic or metaheuristic techniques such as Genetic Algorithms and Tabu Search [1], [12], or
- Allow manual editing but lack intelligent conflict detection and constraint enforcement mechanisms [10].

While advanced optimization approaches including Max-SAT formulations [4], integer programming [3], and hybrid evolutionary models [5] attempt to improve solution optimality, they may introduce

stochastic variability, high computational overhead, or reduced transparency in decision-making. Recent developments incorporating machine learning and adaptive recommendation mechanisms [6], [9] further increase system complexity.

Therefore, there is a need for a constraint-aware and institution-oriented scheduling framework that combines automated generation with controlled, deterministic behavior while continuously enforcing institutional and resource constraints in real time. Such a system must guarantee feasibility, reproducibility, and scalability in multi-branch academic environments.

III. LITERATURE REVIEW

University and high school timetabling is a well-known NP-hard combinatorial optimization problem that requires assigning courses, instructors, rooms, and time slots while satisfying both hard and soft constraints [3]. Over the years, researchers have proposed metaheuristic, exact, hybrid, and intelligent approaches to address its inherent complexity [7]. The difficulty arises from the exponential growth of feasible combinations as the number of resources increases, making exhaustive search computationally infeasible for real-world institutional settings.

Genetic Algorithms (GAs) have proven highly capable of handling the combinatorial nature of timetabling problems. GAs exploit biological concepts such as natural selection, mutation, and crossover to evolve feasible schedules within highly constrained environments [12]. Dave et al. [12] demonstrated the effectiveness of GA-based timetable generation in handling room allocation, faculty assignment, and time-slot optimization. Similarly, evolutionary adaptation mechanisms allow dynamic encoding of schedules to accommodate institutional requirements such as faculty preferences and course structures. The flexibility of GA-based models enables adaptive fitness functions that balance constraint satisfaction with optimization objectives, which is also aligned with personalization strategies discussed in [8]. Despite their flexibility, GA-based systems may sometimes generate intermediate infeasible solutions

that require repair mechanisms to enforce strict institutional constraints.

To overcome such limitations, Constraint Satisfaction Models (CSMs) and exact optimization approaches have been explored. Friedman [3] formulated timetabling as a large-scale integer programming problem, ensuring strict adherence to institutional constraints. Likewise, Ceschia et al. [7] provided a comprehensive overview of benchmark datasets and constraint modeling strategies, emphasizing preprocessing techniques that eliminate infeasible allocations early in the search process. Such preprocessing significantly reduces computational complexity and improves solution quality. Exact methods provide guaranteed feasibility but may suffer from scalability issues when applied to large academic institutions with multiple departments and shared resources.

Hybrid approaches combining evolutionary algorithms with constraint-based reasoning have demonstrated superior performance in balancing feasibility and optimization. Hooshmand et al. [1] proposed a tabu search algorithm with diversification strategies to escape local optima, thereby improving exploration of feasible solution spaces. El Halaby [4] transformed timetabling constraints into a Max-SAT formulation, ensuring strict logical consistency in assignments. Similarly, Lalitha et al. [10] developed a web-based timetable generator integrating conflict resolution algorithms, showing practical implementation of hybrid scheduling techniques suitable for institutional deployment.

Recent research trends emphasize intelligent and adaptive scheduling mechanisms. Kenekayoro [6] incorporated machine learning techniques to evaluate and refine timetable solutions, introducing data-driven decision support for improved timetable quality. Wu et al. [9] proposed a digital twin driven recommendation framework for adaptive campus timetabling, enabling dynamic adjustments based on real time constraints and environmental feedback. Additionally, Králev et al. [2] introduced event grouping strategies to reduce computational overhead by clustering compatible scheduling entities before allocation, thereby improving scalability.

Metaheuristic diversification strategies such as tabu search [1] and optimization under minimal perturbation constraints [5] further enhance robustness in real world scenarios where schedule modifications are inevitable. These approaches aim to balance exploration and exploitation while maintaining feasibility under evolving institutional requirements. Moreover, several studies highlight the importance of combining strict hard-constraint enforcement with soft-constraint optimization to produce high-quality schedules that are both practical and adaptable [7].

Overall, the literature indicates that while GAs provide flexibility and adaptability, constraint based and exact models ensure strict feasibility. Hybrid systems that combine evolutionary search, constraint modeling, preprocessing strategies, and intelligent evaluation mechanisms demonstrate improved performance in generating optimized, conflict free schedules [1], [7]. However, many existing approaches lack efficient handling of parallel resource allocation, consecutive laboratory block enforcement, and independent batch scheduling under dynamic academic configurations. These limitations motivate the development of a deterministic rule based and constraint-aware scheduling framework capable of ensuring strict feasibility while supporting adaptive, real time timetable generation.

IV. SUGGESTED FRAMEWORK

The proposed framework adopts a Deterministic RuleBased Scheduling Approach for solving the university course timetabling problem. Unlike evolutionary and stochastic approaches such as Genetic Algorithms (GAs) [1], [12] and hybrid metaheuristics [5], the proposed system relies on predefined scheduling rules and constraint-driven allocation to generate consistent and reproducible timetables.

Educational timetabling is a well-known NP-hard combinatorial optimization problem, as discussed in [7]. Various approaches including Tabu Search [1], Integer Programming [3], Max-SAT formulations [4], hybrid evolutionary models [5], and machine learning-assisted evaluation [6] have been proposed in literature. While these approaches provide strong

optimization capabilities, they often involve high computational complexity and stochastic variability in outputs.

In contrast, the proposed deterministic rule-based framework emphasizes strict enforcement of hard constraints during the scheduling process. Similar to Constraint Satisfaction Models (CSMs) discussed in [3], [7], the system ensures that non-negotiable constraints such as teacher availability, room occupancy, batch conflicts, and laboratory capacity are satisfied at every step of allocation.

The framework is organized into multiple functional layers, including input configuration, constraint modeling, scheduling logic, validation mechanisms, and output visualization. Each layer operates cohesively to transform institutional requirements into structured allocation decisions. Allocation follows a prioritized strategy in which resource-intensive academic activities are handled with higher precedence, thereby reducing fragmentation and improving overall coordination of shared resources. By embedding constraint validation within the allocation process itself, the system minimizes the likelihood of downstream conflicts and reduces the need for post-generation corrections.

The scheduling mechanism dynamically evaluates feasibility conditions at each stage of allocation, allowing the system to adapt its decisions based on constraint satisfaction status. Rather than applying global optimization heuristics, the framework emphasizes incremental feasibility assurance, ensuring that each assignment aligns with institutional timing rules, resource availability, and structural academic policies.

Rather than focusing on stochastic optimization, the framework prioritizes transparency, controlled allocation behavior, and practical deployability. This design philosophy enhances interpretability of scheduling decisions and simplifies administrative oversight. Such an approach is particularly suitable for structured academic environments where reliability, clarity of rule enforcement, and institutional compliance are preferred over aggressive search-based optimization techniques.

Overall, the proposed framework integrates principles of constraint satisfaction [3], structured allocation strategies [2], and conflict-aware scheduling [10] into a coherent, scalable, and institution-oriented timetable generation approach.

V. SYSTEM DESIGN AND ARCHITECTURE

A. OVERALL ARCHITECTURE

The Smart Timetable Generator (STTG) is designed using a modular and layered architecture to ensure scalability, maintainability, and reliable timetable generation. The system is structured into four primary layers: the Presentation Layer, the Application Layer, the Data Layer, and the Conflict Validation Layer. The Presentation Layer provides an interactive interface for administrators to configure academic parameters and visualize generated timetables. The Application Layer contains the core scheduling logic and business rules that govern timetable creation. The Data Layer manages persistent storage of institutional data, including branches, subjects, faculty details, room configurations, and scheduling metadata. The Conflict Validation Layer operates in conjunction with the scheduling engine to ensure compliance with institutional constraints during allocation. Unlike stochastic or metaheuristic approaches such as Genetic Algorithms [1], [12], the proposed system follows a deterministic rule-based constraint scheduling methodology. In this approach, hard constraints are strictly enforced during the scheduling process, ensuring consistent and reproducible outputs for identical inputs.

B. SYSTEM MODULES

The system comprises several interconnected modules that collectively support automated timetable generation. An authentication module ensures secure access through administrative login and session management. This prevents unauthorized modifications and maintains system integrity.

The configuration module enables dynamic setup of institutional parameters, including branch selection, subject allocation, faculty mapping, laboratory configuration, room management,

working days, and period structures. This flexibility allows the system to adapt to varying academic structures without requiring code-level changes. All configuration data is stored in a centralized database to support persistent and scalable scheduling.

A resource allocation module manages the association between different resources. Special handling is incorporated for laboratory sessions, which typically involve batch divisions and higher resource constraints. The system determines appropriate allocation strategies based on institutional parameters while maintaining resource feasibility.

C. SCHEDULING STRATEGY

The core of STTG is a deterministic scheduling engine that prioritizes constraint satisfaction. The scheduling process follows a structured allocation sequence in which higher-constrained academic activities are handled before standard lecture sessions. This layered allocation strategy reduces fragmentation and improves feasibility under limited resource conditions.

Throughout the scheduling process, the system continuously validates teacher availability, room occupancy, batch compatibility, and institutional timing constraints. Rather than generating a complete schedule and validating it afterward, constraint checking is embedded within the allocation process itself. This integrated validation mechanism ensures that conflicts are avoided during generation rather than corrected post hoc. While the generation process is guided by predefined rules and constraint validation mechanisms, minor variations in allocation order may occur depending on system state and configuration sequences. However, the system consistently ensures conflict-free outputs that satisfy institutional requirements. This controlled allocation behavior enhances reliability and administrative confidence in practical academic environments.

D. WORKFLOW OVERVIEW

The scheduling workflow begins with administrative configuration of academic data through the user interface. Once institutional parameters are finalized, the scheduling engine retrieves the relevant data from the database and initializes an internal

timetable structure. Allocation decisions are then made according to predefined constraint rules and institutional priorities. Upon successful allocation, the final timetable is rendered in structured grid form for branch-wise and consolidated master views.

E. USER AND DATA FLOW

The system follows an administrator-driven interaction model. Authorized users configure academic inputs, initiate timetable generation, and review the generated schedule through visualization modules. The data flow follows a structured sequence in which input data is stored in the database, retrieved by the scheduling engine, processed under constraint validation mechanisms, and rendered for output visualization. The database functions as the central repository for all scheduling entities, ensuring consistency across branches and shared institutional resources.

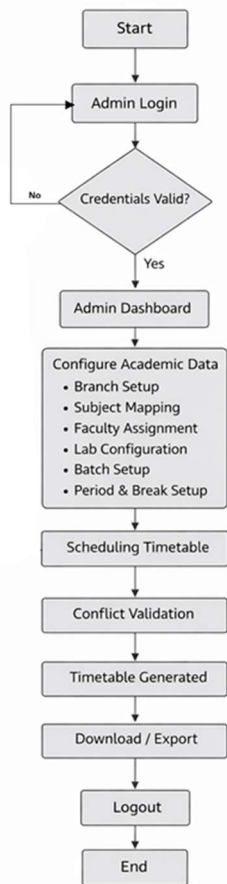


Fig. 1. User Flow Diagram of the Smart Timetable Generator

F. DESIGN CHARACTERISTICS

The Smart Timetable Generator emphasizes deterministic execution, strict enforcement of hard constraints, modular branch-based scheduling, and dynamic configuration capabilities. The architecture prioritizes reliability, predictability, and maintainability while ensuring compliance with institutional policies. Unlike metaheuristic-heavy approaches, the proposed framework focuses on transparency and controlled allocation, making it suitable for structured academic environments.

Fig. 2 Configuration Page

USHA MITTAL INSTITUTE OF TECHNOLOGY									
ACADEMIC YEAR 2025-27 W.E.F 15/07/2026									
EVEN Semester CLASS: AI1									
Day	1	2	3	4	LUNCH BREAK	5	6	7	8
Mo	B1 - CD LAB B2 - AI LAB B3 - BEE LAB B4 - CCH LAB	VK - IC Lab NS - NS PMK - NPTEL Lab SI - NPTEL	AI 011	SS	Lunch				
Tu	B1 - AI LAB B2 - BEE LAB B3 - CCH LAB B4 - CD LAB	PK - NPTEL PV - AI SI - NPTEL VK - IC Lab	ATC 011	NS	Lunch				
We	BEE 011	SSH 011	ATC 011	NS AI 011	SS	Lunch			
Th	ATC 011	NS 011	BEE 011	SSH		Lunch			
Fr	AI 011	SS 011	SSH		Lunch	B1 - CCH LAB B2 - CD LAB B3 - AI LAB B4 - BEE LAB	SI - NPTEL VK - IC Lab PS - CD AP - Drawing Hall	B1 - BEE LAB B2 - CCH LAB B3 - CD LAB B4 - AI LAB	RK - AC Lab SI - NPTEL VK - IC Lab NS - ASLT

Teachers: AI1 - Dr. Anil Kumar, SS - Dr. Seema Handhate, SSH - Dr. Sushma Shrivastava, ATC - Dr. Anil Kumar, NS - Dr. Nirmala Shrivastava, PK - Mr. Prakash Khadke, PMK - Mr. Prakash Mahapatra, PV - Mr. Poojaram V., SI - Dr. Sushma Shrivastava, VK - Dr. Vinay Khare.

Classrooms / Labs: 011 - 011, ACL Lab - Advance Communication Lab, AEL - Advance EMI Lab, AGL - Applied Science Lab I, CCL - Control & Instrumentation Lab, Drawing Hall - Drawing Hall, IC Lab - Integrated Circuits Lab, MPEL - Microprocessor Lab, New2 - New2, NPTEL Lab - NPTEL Lab.

Subjects / Labs: AI - AI, ATC - ATC, BEE - BEE, Lab - AI LAB, BEE LAB, CCH LAB, CD LAB.

Time Table Incharge: Principal, Class Coordinator

Fig. 3 Generated Branch-wise Timetable

Lab Room:

LAB ROOM: Integrated Circuits Lab									
Day	1	2	3	4	LUNCH	5	6	7	8
Mo	AI1 - B1 - CD LAB (Dr. Vinay Khare)				Lunch				
Tu	AI1 - B4 - CD LAB (Dr. Vinay Khare)				Lunch				
We					Lunch				
Th					Lunch				
Fr					Lunch	AI1 - B2 - CD LAB (Dr. Vinay Khare)		AI1 - B3 - CD LAB (Dr. Vinay Khare)	

Shows: Class/Room/Lab

Fig. 4 Lab/Lecture Room View

MASTER TIMETABLE VIEW (All Branches)																																								
Branch	Monday							Tuesday							Wednesday							Thursday							Friday											
	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7					
AI1	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI
ITZ	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI
AI2	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI	AI

Selected Teacher: Dr. Seema Handhate

Fig. 5 Master View of the Smart Timetable Generator

VI. OBSERVATION

The Smart Timetable Generator emphasizes structured constraint enforcement, modular branch-based scheduling, and prioritized allocation of resource-intensive academic activities. The framework is designed to support dynamic academic configuration while maintaining strict adherence to

institutional policies such as faculty availability, room occupancy, and laboratory continuity requirements. By integrating rule-driven allocation with continuous constraint validation, the system ensures the generation of feasible and conflict-free timetables across multiple departments.

The architecture prioritizes reliability, maintainability, and institutional compliance over aggressive optimization techniques. While the allocation process follows predefined scheduling rules, it remains flexible enough to adapt to varying academic configurations. This controlled and constraint-aware approach aligns with structured timetabling models discussed in prior research [3,7], while differing from heuristic-intensive and stochastic metaheuristic methods [1,2] that prioritize search-based optimization.

VII. ACTION PLAN/FUTURE WORK

Future enhancements may include integrating machine learning techniques to analyze historical faculty availability and room usage patterns for predictive and more optimized timetable generation. Machine learning-based predictive modeling has been explored in educational timetabling research to enhance adaptability and decision support [7], [12]. This can improve scheduling efficiency, reduce potential conflicts, and support data-driven decision-making.

The system can also be extended with analytical dashboards for workload and resource utilization insights, similar to data-driven institutional scheduling systems discussed in web-based academic management frameworks [10]. Additionally, multi-department scalability and advanced academic planning features can further strengthen its adaptability and institutional applicability, as highlighted in large-scale university timetabling studies [3], [7].

VIII. CONCLUSION

The Smart Timetable Generator (STTG) presents a deterministic, rule-based solution to the complex problem of academic timetable scheduling. Unlike probabilistic or evolutionary approaches, the

proposed system enforces strict hard constraints during allocation, ensuring conflict-free scheduling through a structured strategy, overall significantly reducing manual intervention, improves transparency, and enhancing resource utilization across departments.

The implementation demonstrates the practical applicability of the system in a real institutional environment. The proposed solution provides a scalable, dynamic, maintainable, and predictable framework for timetable generation, making it suitable for deployment across multiple departments and academic structures.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to our mentor and project supervisor, Prof. Sujata Kullur, for her invaluable guidance, continuous support, and constructive feedback throughout the development of our research on the Smart Timetable Generator. Her expertise, encouragement, and insightful suggestions played a crucial role in shaping and refining this work.

We are also thankful to the Department of Information Technology, Usha Mittal Institute of Technology, for providing the necessary resources and a supportive academic environment that enabled us to successfully complete this project.

In addition, we extend our appreciation to the researchers and scholars whose published works have contributed significantly to our understanding and helped guide our study

REFERENCES

- [1] S. Hooshmand, M. Behshameh, and O. Hamidi, "A tabu search algorithm with efficient diversification strategy for high school timetabling problem," unpublished.
- [2] V. Kraliev, R. Kralieva, and B. Yurukov, "An event grouping based algorithm for university course timetabling problem," unpublished.
- [3] J. S. Friedman, "Automated timetabling for small colleges and high schools using huge integer programs," June 22, 2021.
- [4] M. El Halaby, "Solving the course-timetabling problem of Cairo University using Max-SAT," unpublished.
- [5] J. Kotas, S. Koellmann, and P. Pham, "Optimal minimal-perturbation university timetabling with faculty preferences," arXiv:2008.12342v1 [cs.AI], Aug. 31, 2020.
- [6] P. Kenekayoro, "Incorporating machine learning to evaluate solutions to the university course timetabling problem," unpublished.

- [7] S. Ceschia, L. Di Gaspero, and A. Schaerf, "Educational timetabling: Problems, benchmarks, and state-of-the-art results," preprint.
- [8] A. Krystallidis and R. Ruiz-Torrubiano, "Introducing individuality into students' high school timetables," unpublished.
- [9] K. Wu, X. Ye, S. Jamonnak, and X. Feng, "A digital twin-driven recommendation system for adaptive campus course timetabling," unpublished.
- [10] V. Lalitha, M. K., and G. Youkesh, "Automating timetable generation with conflict resolution algorithms in web-based systems for educational institutions," unpublished.
- [11] T. Mamatha, S. Anvitha, G. V. Aditya, S. Meghana, and P. Veekshitha, "Optischedule algorithm for automatic time table generator," unpublished.
- [12] R. Dave, D. Pandey, T. Jagasia, A. Nagmote, and S. Jhaveri, "Time table generator: Using genetic algorithm," unpublished.