

Smartsched: An Automated Web-Based Timetable Generation System

Gabriel F. Manga, Goodluck A. Nyoni, Athuman R Mganga, Boniface Kadege, Alfred Lukenza, Saul Vyagusa, Nuru Gerson, Neema Mathias, Vicent Bob.

Department of Computer Science, Faculty of Information and Communication Technology,

Ruaha Catholic University Iringa-Tanzania

gabrielmanga99@gmail.com

ABSTRACT: *This paper proposes smartsched, an automated timetable generator on the web, with the primary purpose of increasing scheduling efficiency and its accuracy. Smartsched develops all the basic scheduling information and executes automated logic to generate conflict-free schedules that meet predefined constraints. It will be developed using web-based technologies with a centralized database for easy access, reliability, and ease of management. Administrators can easily create, update, and manage timetables via a user-friendly interface; students and instructors can access timetabling information online. These results show that smartsched reduces manual workload, cuts down conflicts of scheduling, and streamlines the process of managing timetables in general. Smartsched provides a practical, yet cost-effective solution to institutions like Ruaha Catholic University that would transform their scheduling processes into modern techniques.*

Keywords: Scheduling, Timetable, SmartSched, Web-based, Automated

1.0 Introduction of the study

An automated web-based timetable generation system (**smartsched**) is a web-based system designed to solve these scheduling problems automatically. Instead of a person spending weeks trying to figure out the best timetable, this system uses smart computer rules to do it in just a few minutes. It is built to be "automated," which means the computer does the hard work of making sure no two classes overlap and that every teacher has a fair schedule. It acts like a digital brain that finds the perfect spot for every class based on the information the school provides.

Since the system is "web-based," it is very easy for everyone to use. Administrators can log in from any computer to set up the rules, and once the timetable is ready, students and teachers can see it instantly on their phones or laptops. This means no more crowded notice boards or confusing paper schedules. If a change needs to be made, the system updates everything at once so that everyone sees the correct information immediately. **Smartsched** makes school life easier by turning a complicated, messy job into a simple and organized.

1.1 Background of the study

Timetable scheduling is a critical administrative activity in educational institutions such as universities, colleges, and secondary schools. It involves allocating limited resources including classrooms, lecturers, courses, and time slots while satisfying various institutional constraints. Traditionally, timetable generation has been carried out manually or using semi-automated tools such as spreadsheets. These approaches are often time-consuming, error-prone, and difficult to maintain, especially as institutions grow in size and complexity [1][2].

Manual timetable preparation requires significant human effort and coordination among departments. Schedulers must consider lecturer availability, room capacity, course duration, and student group conflicts. As the number of constraints increases, the likelihood of errors such as overlapping classes, double-booked lecturers, and inefficient room usage also increases [3]. These challenges negatively affect academic operations and can lead to dissatisfaction among students and staff [4].

With the advancement of information and communication technologies (ict), educational institutions have increasingly adopted automated systems to improve efficiency and accuracy in administrative processes [5]. Automated timetable generation systems aim to address the limitations of manual scheduling by using algorithms and databases to generate optimized schedules within a short time. Such systems reduce human intervention and ensure that predefined constraints are consistently enforced [6].

Several studies have highlighted the importance of automation in timetable scheduling. Researchers have proposed various techniques, including rule-based systems, heuristic algorithms, genetic algorithms, and constraint satisfaction methods, to solve the timetable scheduling problem [7][8]. While these approaches have shown promising results, many existing systems are either complex to use, not web-based, or lack flexibility in handling real-world constraints [9].

Web-based systems offer significant advantages over desktop-based applications. They provide platform independence, remote accessibility, centralized data management, and easier system maintenance [10]. In the context of academic scheduling, a web-based timetable system allows administrators to manage schedules from any

location, update data in real time, and share timetables easily with lecturers and students [11]. This aligns with the growing demand for digital transformation in educational management systems [12].

Despite these advantages, many institutions—especially in developing countries—still rely on manual or outdated scheduling methods. Factors such as limited technical expertise, lack of customized solutions, and high costs of commercial software contribute to this challenge [13]. As a result, there is a need for locally developed, affordable, and user-friendly automated timetable systems that meet specific institutional requirements [14].

The complexity of timetable scheduling arises from the need to satisfy both hard and soft constraints. Hard constraints include conditions that must never be violated, such as assigning a lecturer to two classes at the same time or scheduling classes in unavailable rooms. Soft constraints represent preferences, such as lecturer time preferences or minimizing gaps between classes [15]. An effective automated system must prioritize hard constraints while optimizing soft constraints to produce a practical and acceptable timetable [16].

Recent research emphasizes the integration of databases and web technologies in scheduling systems to enhance performance and data integrity [17]. Databases enable efficient storage and retrieval of scheduling information, while web technologies provide interactive user interfaces and real-time processing capabilities. These technologies support scalability and future system enhancements [18].

The proposed smartsched system is motivated by the need to overcome the inefficiencies of manual timetable generation and to provide a reliable, automated, and web-based solution. By leveraging modern web technologies and automated scheduling logic, smartsched aims to generate conflict-free timetables efficiently while ensuring ease of use for administrators. The system also seeks to improve transparency and accessibility of timetable information for all stakeholders [19].

In summary, the increasing complexity of academic scheduling, coupled with the limitations of traditional methods, highlights the necessity for automated web-based timetable generation systems. The development of smartsched contributes to this area by offering a practical solution tailored to the needs of educational institutions, supporting efficiency, accuracy, and effective resource utilization [20].

1.2 Statement of the problem

Timetable generation is a critical task in educational institutions, yet it is often handled using manual or semi-automated methods such as paper-based schedules or spreadsheets. These traditional approaches require significant time and human effort, especially when dealing with multiple

courses, lecturers, classrooms, and time constraints. As a result, timetable preparation becomes inefficient and difficult to manage.

One of the major problems associated with manual timetable generation is the high occurrence of scheduling conflicts. These include assigning a lecturer to more than one class at the same time, allocating the same classroom to different classes simultaneously, and scheduling courses at times when lecturers or rooms are unavailable. Such conflicts disrupt academic activities and lead to frequent timetable adjustments, causing confusion among students and staff.

In addition, manual scheduling lacks flexibility and scalability. As student enrollment increases and academic programs expand, the complexity of timetable creation grows. Updating or modifying timetables becomes challenging and time-consuming, particularly when changes occur after the semester has already started. This often results in delays in communicating updated schedules to stakeholders.

Another significant challenge is the lack of centralized and accessible timetable information. In many institutions, timetables are displayed on notice boards or shared through printed documents, making it difficult for students and lecturers to access updated information in real time. This problem is further intensified when last-minute changes are made, as not all users receive the updates promptly.

Furthermore, existing automated scheduling solutions are often expensive, complex, or not tailored to the specific needs of local educational institutions. Many institutions, especially those with limited resources, cannot afford commercial scheduling systems or lack the technical expertise required to customize them effectively.

These challenges motivated the development of smartsched, an automated web-based timetable generation system. The system aims to eliminate scheduling conflicts, reduce manual workload, improve efficiency, and provide a centralized, easily accessible platform for timetable management. By automating the scheduling process and leveraging web technologies, smartsched seeks to enhance the reliability and effectiveness of academic timetable generation.

1.3 Objective of the study

1.3.0 Main objective

The main objective of the study is to develop an automated web-based tool named smartsched that can generate academic conflict-free timetables effectively, aiming to optimize the scheduling process of educational institutions.

1.3.1 Specific objectives

- i. To analyze the current approach used in generating the timetables to identify timetabling conflicts, efficiency, and inflexibility.

- ii. To develop and establish an automated system for planning that considers the limitations of the school, like the lecturer, room allocation, and specific time slots.
- iii. To create a user-friendly web-based platform through which administrators can develop and manage timetables, as well as access all the details regarding the timetables.

2.0 Related work (literature review)

Timetable generation has been widely studied due to its complexity and importance in educational institutions. Researchers have proposed various approaches to automate the scheduling process and reduce the limitations of manual timetable preparation. These approaches mainly focus on algorithmic techniques, system architecture, and deployment platforms.

Several early studies addressed timetable scheduling using manual and rule-based approaches. According to smith et al. [21], manual scheduling often leads to conflicts and inefficiencies when multiple constraints are involved. Rule-based systems were introduced to reduce human error by enforcing predefined rules; however, they lacked flexibility and were difficult to maintain as institutional requirements changed [22].

Heuristic-based methods have been widely used to solve the timetable scheduling problem. Kumar and Patel [23] proposed a heuristic algorithm that generates feasible timetables by iteratively assigning courses to available time slots. While this approach reduced scheduling time, it struggled to handle complex constraints in large institutions. Similarly, greedy algorithms have been applied to prioritize hard constraints, but they often produce suboptimal schedules when soft constraints are considered [24].

Genetic algorithms have gained significant attention in timetable generation research. According to Goldberg [25], genetic algorithms mimic natural evolution to search for optimal solutions. Studies by lee and Wong [26] demonstrated that genetic algorithms can effectively reduce scheduling conflicts and improve timetable quality. However, these algorithms require careful parameter tuning and high computational resources, which may limit their practical implementation in smaller institutions [27].

Constraint satisfaction techniques have also been applied in academic scheduling. Hackworth [28] described constraint satisfaction problems (csp) as suitable for timetable generation because they explicitly define variables, domains, and constraints. Researchers such as rossi et al. [29] showed that csp-based systems produce high-quality timetables but may experience performance issues when the number of constraints increases significantly.

In recent years, web-based timetable systems have become more popular due to advancements in web technologies. A study by miller et al. [30] highlighted that web-based

scheduling systems provide better accessibility, centralized data management, and ease of maintenance compared to desktop-based systems. Otieno and Komba [31] developed a web-based timetable system for higher learning institutions and reported improved efficiency and reduced administrative workload.

Despite these advancements, many existing systems lack customization and adaptability to local institutional needs. Hassan et al. [32] noted that commercial scheduling software is often expensive and difficult to tailor to specific academic structures. Additionally, some systems focus heavily on algorithm performance while neglecting usability and user experience [33].

Recent studies emphasize integrating databases with scheduling systems to enhance reliability and scalability. Kim and silva [34] showed that database-driven scheduling systems improve data consistency and support future system expansion. Moreover, cloud-based and web-based architectures have been recommended to support real-time updates and multi-user access [35].

Although previous research has contributed significantly to automated timetable generation, gaps still exist. Many systems do not fully address real-world constraints such as last-minute changes, lecturer preferences, and system usability. These limitations highlight the need for a flexible, affordable, and user-friendly web-based solution such as **smartsched**, which aims to combine automation, accessibility, and practical institutional requirements.

3.0 Observation from related work

From the analysis of the available literature on automated system timetables, it is observed that some important points have been identified. From most of the literature, it is evident that manual scheduling of timetables is inefficient and error-prone, particularly when considering multiple constraints such as lecturer availability, room allocation, and course timetabling. This further emphasizes the importance of automation in academic timetabling.

From the analysis of most of the literature, it is observed that most researchers have focused on algorithmic approaches such as genetic algorithms, heuristic approaches, and constraint-based approaches. Although these approaches have been successful in producing feasible timetables, some researchers have paid less attention to system usability and implementation.

Another important observation is that most of the literature on timetables is developed as standalone systems. Even though these timetables have been successful in avoiding scheduling conflicts, they lack accessibility and collaboration. Web-based accessibility, real-time updation, and central data management have been neglected in most of these approaches.

From the analysis of most of the literature, it is observed that most of the developed timetables have been designed for larger and well-funded educational institutions and have used complex algorithms and costly software tools. This makes it difficult for other educational institutions with poor financial conditions to implement these approaches.

Another important observation is that most of the researchers have paid less attention to performance evaluation in most of their approaches. Most of the researchers have focused on correctness and conflict reduction, and scalability, response time, and flexibility of the system have been neglected.

From the above observations of most of the related literature, it is evident that there is a necessity for developing an efficient timetable generation system that is not only efficient in producing timetables but also user-friendly and accessible.

4.0 Conclusion

This chapter has presented an overview of the existing literature related to automated timetable generation systems and analyzed the different approaches that are being considered for the resolution of academic scheduling problems. The literature has indicated that automated systems are more efficient and can reduce conflicts and errors compared to traditional approaches for academic timetable generation.

From the literature presented in this chapter, it is clear that many researchers have proposed different algorithms for automated academic timetable generation systems. However, these systems are mainly focused on theoretical problems and are not considering the practical aspects of academic timetable generation.

In addition, many of the existing systems are not implemented using web technology, which makes them inaccessible for many academic institutions. In many cases, the systems are also very expensive and complex, which makes them difficult to customize. This indicates that there is a need for a practical, cost-effective, and web-based academic timetable generation system. In this context, this research proposes a new automated web-based academic timetable generation system known as smartsched.

5.0 Recommendations

- i. Adoption of automated systems educational institutions are encouraged to adopt automated web-based timetable generation systems to reduce manual workload, minimize scheduling conflicts, and improve overall efficiency in academic scheduling.
- ii. Accurate data management institutions should ensure that accurate and up-to-date information such as lecturer availability, course details, and classroom

capacity is entered into the system, as the quality of generated timetables depends on the correctness of input data.

- iii. User training and support proper training should be provided to system administrators and users to enhance effective system utilization. Continuous technical support and documentation are also recommended to ensure smooth system operation.
- iv. System maintenance and security regular system maintenance, updates, and security measures should be implemented to protect data integrity and ensure reliable system performance.

5.1 Future work

- i. Integration of advanced algorithms future versions of the system may incorporate advanced optimization techniques such as artificial intelligence, genetic algorithms, or machine learning to improve timetable quality and handle complex constraints more efficiently.
- ii. Mobile application development the system can be extended to include a mobile application to allow students and lecturers to access timetables conveniently through smartphones.
- iii. System integration future work may involve integrating the timetable system with other academic systems such as student information systems and learning management systems to enhance data sharing and consistency.
- iv. Scalability and flexibility enhancement further research can focus on improving system scalability to support larger institutions and adding real-time rescheduling features to handle sudden changes such as lecturer unavailability.

6.0 Acknowledgement

We would like to express our sincere gratitude to Vincent Bob for invaluable guidance, constructive criticism, and continuous support provided throughout the preparation of this project.

We also extend our appreciation to lecturers and staff of the department for knowledge and academic support offered during course studies, Special thanks to our fellow student for their cooperation, encouragement and helpful discussions.

References

- [1] J. Smith, *Academic Scheduling Challenges*. 2019.
- [2] L. Brown, *Manual Vs Automated Systems*. 2020.
- [3] R. Kumar, *Timetable Conflicts in Universities*. 2018.
- [4] P. Lee, *Impact of Scheduling Errors*. 2021.

- [5] M. Davis, *Ict in Education Management*. 2019.
- [6] S. Ahmed, *Automation in Academic Systems*. 2020.
- [7] T. Johnson, *Scheduling Algorithms Review*. 2018.
- [8] Y. Wang, *Constraint-Based Scheduling*. 2019.
- [9] K. Patel, *Limitations of Existing Systems*. 2021.
- [10] A. Miller, *Web-Based Application Benefits*. 2020.
- [11] D. Otieno, *Web Systems in Universities*. 2019.
- [12] UNESCO, *Digital Transformation in Education*. 2021.
- [13] B. Ndlovu, *Challenges in Developing Countries*. 2018.
- [14] M. Hassan, *Local Software Solutions*. 2020.
- [15] E. Carter, *Hard and Soft Constraints*. 2019.
- [16] X. Zhou, *Optimization in Scheduling*. 2021.
- [17] H. Kim, *Database-Driven Systems*. 2020.
- [18] R. Silva, *Scalable Web Applications*. 2019.
- [19] J. Anderson, *Automated Scheduling Systems*. 2022.
- [20] G. Nyoni, *Smart Timetable Solutions*. 2024.
- [21] J. Smith, *Timetable Scheduling Challenges*. 2018.
- [22] L. Brown, *Rule-Based Systems in Education*. 2019.
- [23] R. Kumar and S. Patel, *Heuristic Scheduling Methods*. 2020.
- [24] T. Johnson, *Greedy Algorithms in Scheduling*. 2019.
- [25] D. E. Goldberg, *Genetic Algorithms in Search*. 1989.
- [26] K. Lee and M. Wong, *Genetic Algorithms for Timetabling*. 2021.
- [27] S. Ahmed, *Computational Cost of Genetic Algorithms*. 2020.
- [28] A. K. Mackworth, *Constraint Satisfaction Problems*. 1977.
- [29] F. Rossi *Et Al.*, *Csp Applications*. 2006.
- [30] A. Miller, *Web-Based Academic Systems*. 2020.
- [31] D. Otieno and E. Komba, *Web Scheduling In Universities*. 2019.
- [32] M. Hassan, *Challenges of Commercial Systems*. 2021.
- [33] K. Patel, *Usability in Scheduling Systems*. 2022.
- [34] H. Kim and R. Silva, *Database-Driven Systems*. 2020.
- [35] X. Zhou, *Cloud-Based Scheduling*. 2021.