

# Automated University Timetable Generation System using PHP

**Preethi G.<sup>1</sup>, Pratippa T.<sup>2</sup>, Sesa Janani S.<sup>3</sup>**

<sup>1</sup>Assistant Professor, <sup>2,3</sup>Student, Department of Information Technology, Velammal Engineering College, Chennai, India

**E-mail:** <sup>1</sup>preethi.g@velammal.edu.in, <sup>2</sup>pratippathangaraj@gmail.com, <sup>3</sup>saiseshajanani786@gmail.com

## Abstract

The proposed work focuses on preparing the timetable for university departments and scheduling examinations for each semester. Most universities today have multiple departments, each offering different undergraduate and postgraduate programs. To manage all these courses effectively and without any chaos or confusion, a proper timetable or schedule is essential. Proper scheduling, with reserved time slots for each subject, avoids confusion among students and lecturers, preventing overlaps in class timings for the subjects assigned to each teacher. Since manual timetable preparation is very tedious, time-consuming, and prone to conflicts, there is a need for an automatic timetable scheduler that reduces human intervention. To address this need, the proposed method introduces an automatic timetable generator using PHP, MySQL, HTML, and CSS. PHP is used as the programming language, MySQL for the database, and HTML and CSS for the frontend. The system is integrated with Oracle for advanced data management.

**Keywords:** Automatic, Web-Application, PHP, MySQL, HTML, CSS

## 1. Introduction

Timetable generation for scheduling classes and examinations is very stressful and poses many challenges and conflicts when done manually [1]. This happens because there are multiple undergraduate (UG) and postgraduate (PG) programs offered in a particular department, and the number of staff members, classrooms, and lab rooms does not match the

programs available and the student strength [2]. Therefore, there arises a need for an automatic timetable generator that can schedule the time for each subject without any overlap [3]. The proposed work is specifically designed for generating the timetable for a particular department in a college. By providing a platform for students and staff to access the final timetable for a semester, complete with class allocation details, the aim is to revolutionize the scheduling landscape. The proposed system aims to solve the challenges in manual timetable generation by developing a website that automatically generates the timetable with inputs such as student, staff, and subject details [4-10].

### **1.1 Key Features and Scope**

The proposed work aims to develop a website that automates timetable generation, reducing the time taken to create manual timetables. The proposed system can be customized to meet the specific needs of schools and colleges. It aims to provide proper timetable scheduling by addressing obstacles such as limited classroom availability, teacher and student schedules, and subject requirements.

## **2. Related Work**

The author developed a new algorithm using a genetic algorithm (GA) to solve the timetabling problem. This algorithm is faster and easier to use compared to other algorithms and has been tested on real data from multiple high schools and colleges [11]. Most existing research utilizes metaheuristic algorithms and evolutionary computations to handle the complexity and constraints associated with timetable scheduling, offering flexibility in adapting to various constraints and objectives.

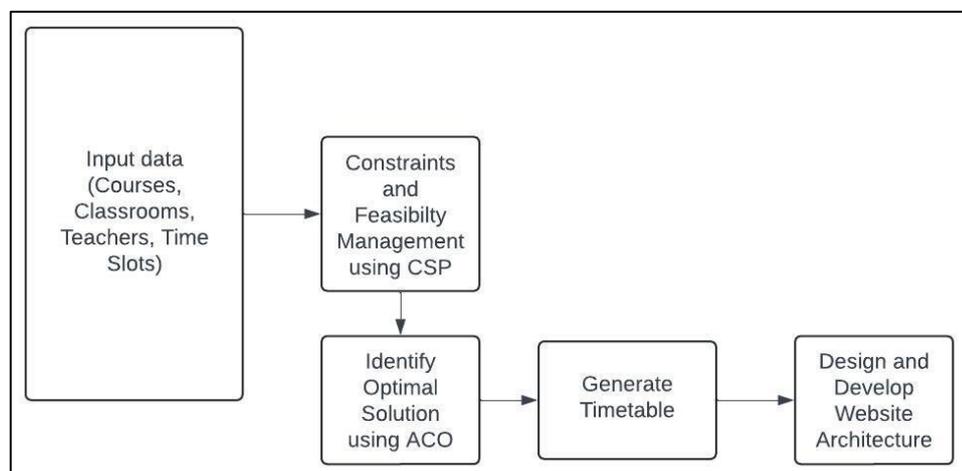
The ability of these algorithms to explore complex solution spaces, manage constraints, and adapt to various requirements makes them valuable tools for timetable scheduling [12-23]. This study demonstrates how neural network algorithms can be used to schedule classes in an educational setting. Graph coloring or graph partitioning, which is classified as an NP-complete problem, presents challenges in scheduling classes.

The study proposes neural network algorithms to address these problems, adopting them to create timetables that can be implemented on digital or analog neural networks [8]. This research provides a solution for creating timetables for students, lecturers, and classrooms,

grouping students for large courses that occur multiple times during the week. Entities are assigned to resources through penalty terms for conflicts and excessive classroom use, with heuristic procedures formulated as assignment-type problems [19]. An online model is presented in this article to choose the most appropriate solution based on intelligent exploration of the solution space [20]. Based on the literature studied, the system aims to develop an automated university timetable generation system using PHP for programming, MySQL for the database, and HTML and CSS for the frontend. The system will be further integrated with Oracle for advanced data management.

### 3. Proposed Work

The proposed work is designed to address the challenges associated with manually scheduling timetables by developing a website that automatically generates timetables for managing different subjects offered in UG and PG programs, along with the available staff and rooms. Currently, the website is developed to generate the timetable for a particular department and can be customized in the future to meet the needs of all departments in the university. Figure 1 below shows the overall workflow of the proposed system.



**Figure 1.** Proposed Workflow

As stated in Figure 1, input data such as course names, durations, classroom names, capacities, teacher names, subjects they can teach, their availability, and the available time slots for scheduling classes are gathered. Constraints such as avoiding double-booking of classrooms, time slot conflicts, and teacher availability are specified. The Constraint

Satisfaction Problem (CSP) [25] model is used to organize and address these constraints effectively with a feasible solution. Once the constraints are managed using CSP, the scheduling with the optimal solution is carried out using Ant Colony Optimization (ACO) [24]. Integrating CSP and ACO ensures feasibility and optimizes the timetable generation. The Figure 2 shows the algorithm for timetable scheduling.

```

User Input
{
//Initialize Input
{
subject, classrooms, staff name, timeslots
}
//Initialize constraints
{
no double-booking, teacher availability, student enrollment, timeslots
}
//Apply CSP
For
assignment (subject name, staff name, classroom)
{
If (Classroom booked or Staff unavailable)
{
Reject Assignment
}
Else {
Assign (Subject, Classroom, Staff, TimeSlot)
}
}
//Initialize ACO
Initialize ACO (Ants, PheromoneLevels)
For each Ant
{
Construct Timetable
If Feasible {
Evaluate Solution Quality
}
Update Pheromone Levels
Repeat until convergence
}
}

```

**Figure 2.** Algorithm

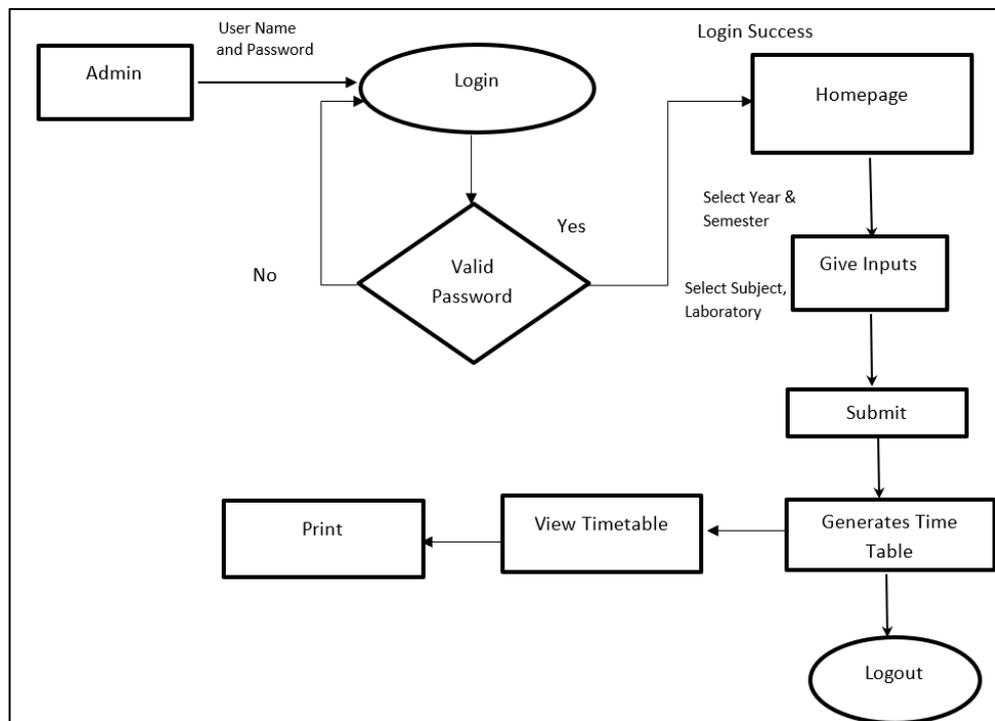
The following details shows the details of the software used for developing the proposed website

1. **PHP:** Is used in the blackened for implementing CSP and ACO algorithms and interacting with Oracle.
2. **MYSQL:** MySQL is used as a database to store and manage data related to the university timetabling problem. It can store information such as course schedules, classroom availability, faculty schedules, enrollments, and the overall timetable.

**3.HTML:** HyperText Markup Language (HTML) is utilized for creating the structure and content displayed on the frontend. It is used to collect inputs from users, enabling the system to perform its operations.

**4.CSS:** Cascading Style Sheets (CSS) is used for styling the content in HTML. It works alongside HTML to improve the visual appearance, making the content more attractive and user-friendly. CSS enhances the overall user interface by providing various styling options.

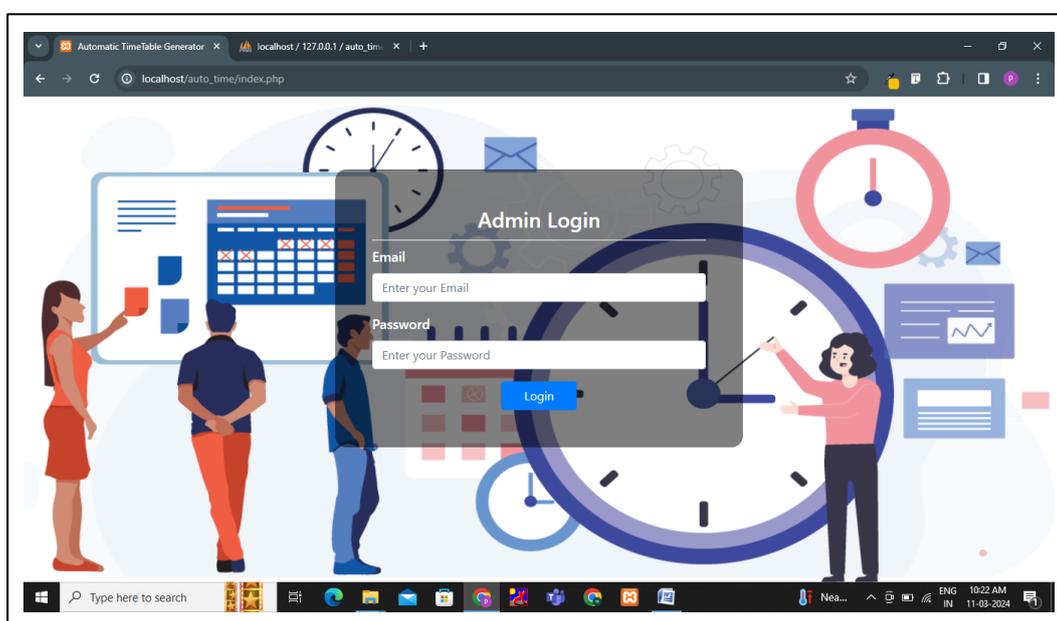
In this work, a prototype has been developed using HTML and CSS for the frontend and connected to a MySQL database. The system prioritizes the first three subjects provided as input and schedules labs twice a week, with an option for a separate timetable for Saturdays. The prototype system addresses university timetabling for both examinations and courses using heuristic algorithm and mathematical approach. It features an interactive Windows-based graphical user interface (GUI) that allows users to interact with the system easily and efficiently, helping them to find and achieve satisfactory results. The flowchart below in figure 3 shows the working of the website.



**Figure 3.** Data Flow Diagram

#### 4. Results and Discussion

The Automated Timetable Generator (ATG), was developed defining the requirements, including the necessary features and constraints. The system architecture was designed, outlining the frontend, backend, and database structure. The frontend was developed by creating the user interface using HTML, CSS, and JavaScript. The backend was implemented with PHP scripts to handle data and execute algorithms. The database schema was designed to manage and store data effectively. The CSP and ACO algorithms were developed to optimize timetable generation. Finally, all components were integrated by connecting the frontend, backend, and database to ensure seamless functionality. The Figure 4 shows the admin and the login page.



**Figure 4.** Admin and Login Page

The administrator is the only user who can log into the website and enter the necessary information. Login credentials are saved on the data server. If the provided username and password are correct, the system will verify them and proceed to load the webpage.

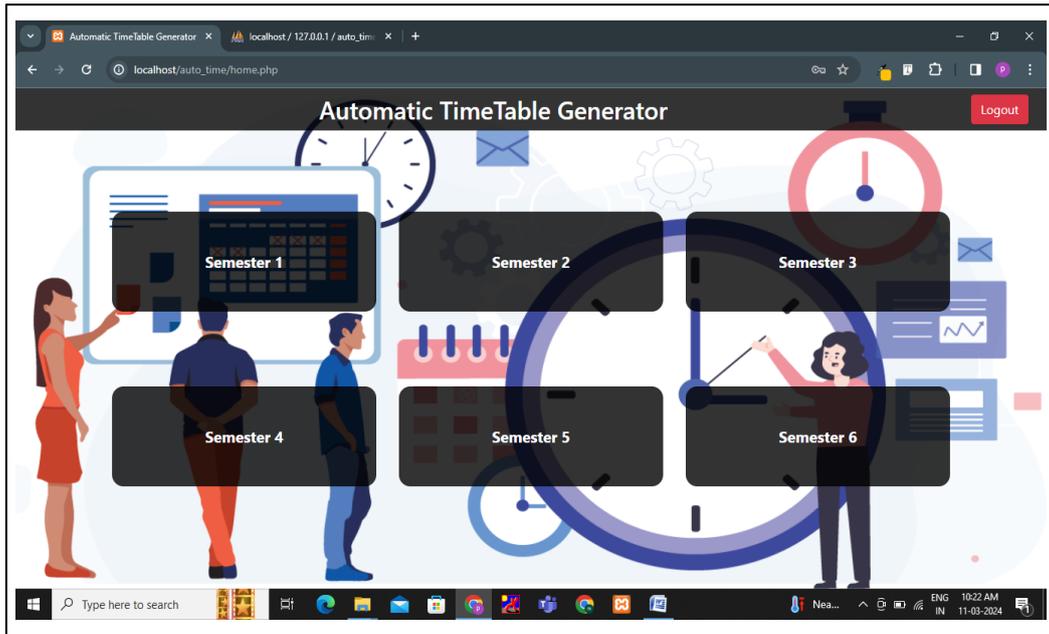


Figure 5. Home Page

The admin can select the necessary fields and enter data, including choosing the semester and year, using the navigation bars with various options displayed on the home page (Figure 5). There are up to 6 semesters available for selection, allowing the admin to choose the specific class for which the timetable should be generated.

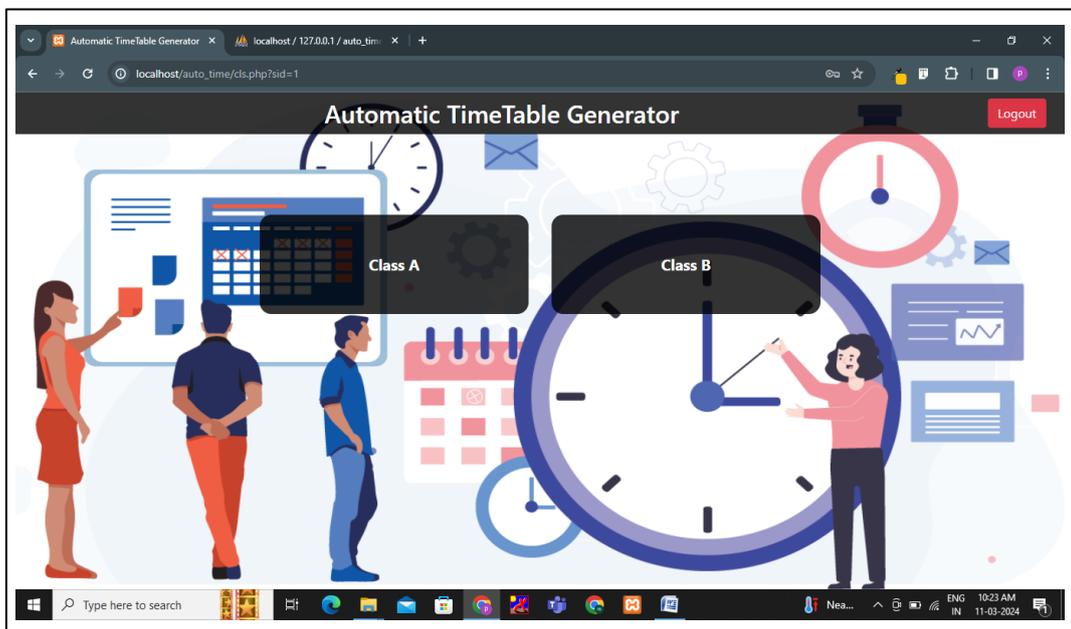


Figure 6. Section Selection Page

In this Figure 6, it shows how users can specifically select the section under a particular semester to generate the timetable using the provided inputs with PHP code.

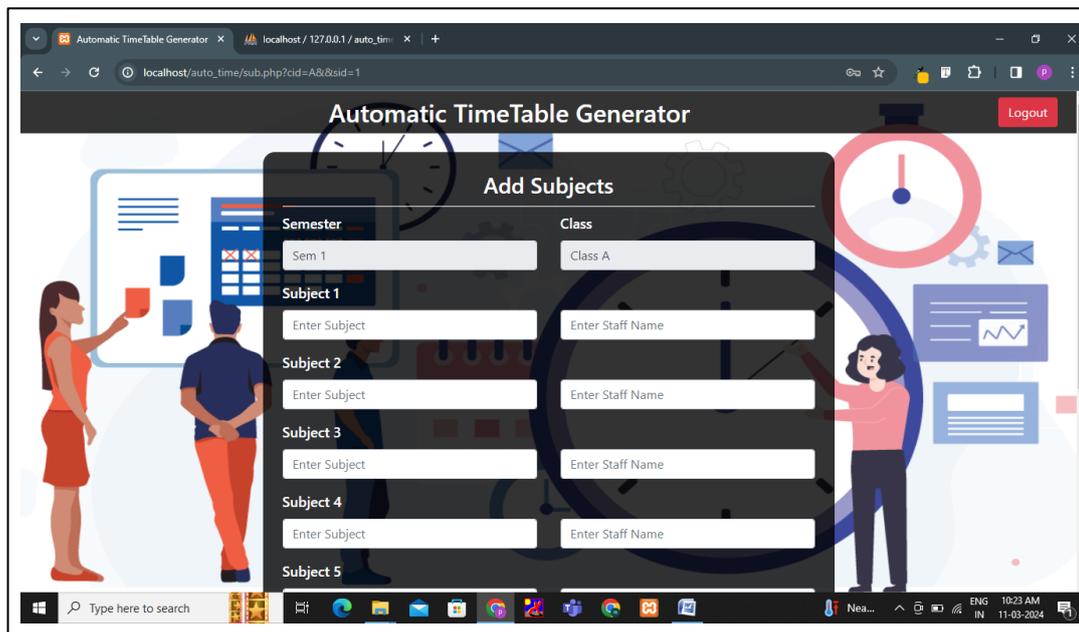


Figure 7. Input Page

The details page in Figure 7 includes various fields, such as subjects and laboratories, to ensure that the system understands the administrators' needs and generates a timetable that avoids conflicts.

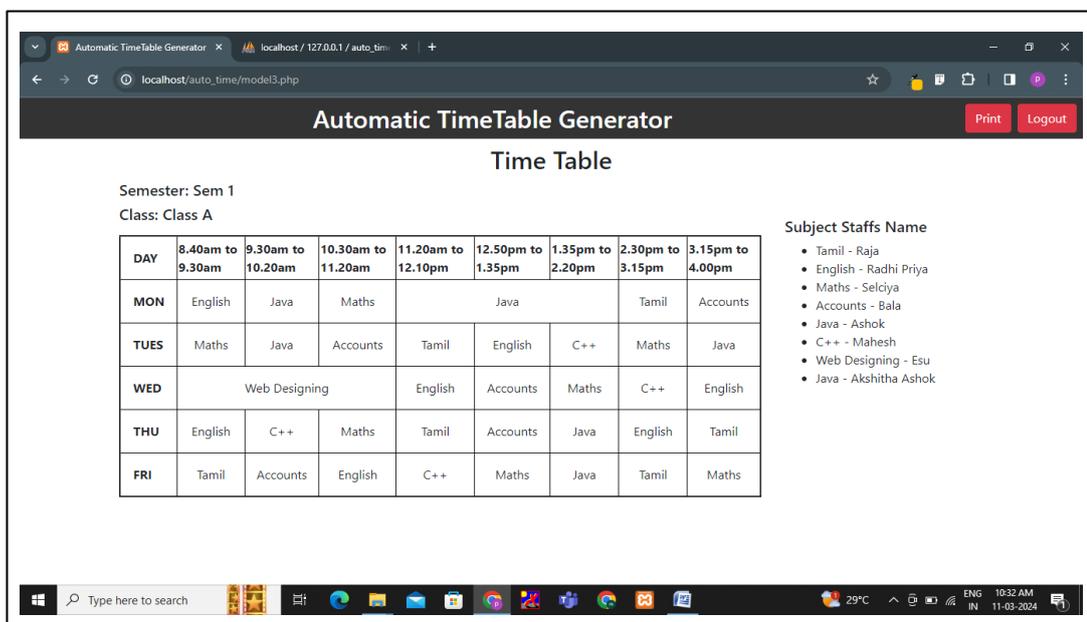


Figure 8. Generated Timetable Page

The system automatically generates a schedule with the staff members appropriately assigned and with enough classes for a week based on the admin's provided details as shown in Figure 8. This uses the PHP code to generate timetable with some constraints. The generated page can be printed for further use.

## 5. Conclusion and Future Work

The Automated Timetable Generator (ATG) aims to streamline educational scheduling by providing an efficient and user-friendly solution. By inputting subjects, faculty names, preferences, and constraints, the system uses heuristic algorithm and mathematical approach to generate optimized timetables, considering resource availability and minimizing conflicts. Key features include flexible input options, conflict resolution mechanisms, optimization techniques, adaptability to real-time changes, and intuitive visualization. Benefits include time savings, improved accuracy, enhanced efficiency, increased satisfaction, and scalability for institutions of all sizes. Overall, the ATG ensures optimal resource utilization, minimal conflicts, and a better learning experience for students and faculty. Future enhancements for the automatic timetable generator will include a machine learning-driven optimization engine, a mobile-friendly interface, real-time updates, and customizable reporting. These improvements will make the system more efficient, adaptive, and collaborative, while ensuring robust security and a user-friendly experience.

## Reference

- [1] Bhaduri, Antariksha. "University time table scheduling using genetic artificial immune network." In 2009 International Conference on Advances in Recent Technologies in Communication and Computing, Kottayam, Kerala. IEEE, 2009. 289-292.
- [2] Srinivasan, Dipti, Tian Hou Seow, and Jian Xin Xu. "Automated time table generation using multiple context reasoning for university modules." In Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 Honolulu, HI, USA (Cat. No. 02TH8600), vol. 2, IEEE, 2002. 1751-1756.

- [3] Chowdhary, Anuja, Priyanka Kakde, Shruti Dhoke, Sonali Ingle, Rupal Rushiya, and Dinesh Gawande. "Timetable generation system." *International Journal of Computer Science and Mobile Computing* 3, no. 2 (2014): 410-414.
- [4] Nanda, Anirudha, Manisha P. Pai, and Abhijeet Gole. "An algorithm to automatically generate schedule for school lectures using a heuristic approach." *International journal of machine learning and computing* 2, no. 4 (2012): 492-495.
- [5] Elkhyari, Abdallah, Christelle Guéret, and Narendra Jussien. "Solving dynamic timetabling problems as dynamic resource constrained project scheduling problems using new constraint programming tools." In *Practice and Theory of Automated Timetabling IV*, Springer-Verlag, 2003. 39-59.
- [6] Carter, Michael W., and Gilbert Laporte. "Recent developments in practical course timetabling." In *Practice and Theory of Automated Timetabling II: Second International Conference, PATAT'97 Toronto, Canada, August 20–22, 1997 Selected Papers 2*, Springer Berlin Heidelberg, 1998. 3-19.
- [7] Schaerf, Andrea. "A survey of automated timetabling." *Artificial intelligence review* 13 (1999): 87-127.
- [8] Neufeld, G. A., and John Tartar. "Graph coloring conditions for the existence of solutions to the timetable problem." *Communications of the ACM* 17, no. 8 (1974): 450-453.
- [9] Mrunal Gaikwad, Adwait Gaikwad, Mukesh Chaudhary., "Auto Timetable Generator", *International Research Journal of Modernization in Engineering Technology and Science* e-ISSN: 2582-5208 vol 4, issue 5 May 2022. 2976-2981
- [10] Farzana Jawale, Renuka Sabale, Dhanashri Kulkarni., "Automated Timetable generator System", *International Research Journal of Modernization in Engineering Technology and Science* e-ISSN: 2582-5208, vol-03,issue 06, 326 -330.June 2021.
- [11] Shraddha Thakare, Tejal Nikram; *Automated Timetable Generator using Genetic Algorithm*, Vol. 9 Issue 07, July 2020. 1425-1427

- [12] Dowsland, Kathryn A. "Simulated annealing solutions for multi-objective scheduling and timetabling." *Modern heuristic search methods* (1996): 155-166.
- [13] Hertz A., "Tabu search for large scale timetabling problems", *European Journal of Operations Research*, vol-54, 1991, 39-47.
- [14] Abbas, Abdulwahed M., and Edward PK Tsang. "Constraint-based timetabling-a case study." In *Proceedings ACS/IEEE International Conference on Computer Systems and Applications*, IEEE, 2001. 67-72.
- [15] Di Stefano, Calogero, and Andrea GB Tettamanzi. "An evolutionary algorithm for solving the school time-tabling problem." In *Workshops on Applications of Evolutionary Computation*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. 452-462.
- [16] Terashima-Marin, Hugo, Peter Ross, and Manuel Valenzuela-Rendon. "Clique-based crossover for solving the timetabling problem with GAs." In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99* (Cat. No. 99TH8406), vol. 2, IEEE, 1999. 1200-1206.
- [17] Paechter, Ben, Andrew Cumming, Henri Luchian, and Mihai Petriuc. "Two solutions to the general timetable problem using evolutionary methods." In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, IEEE, 1994. 300-305.
- [18] Caldeira, J. P., and Agostinho C. Rosa. "School timetabling using genetic search." *Practice and Theory of Automated Timetabling*, Toronto (1997).
- [19] Carter, M.W., Laporte, G. (1998). Recent developments in practical course timetabling. In: Burke, E., Carter, M. (eds) *Practice and Theory of Automated Timetabling II. PATAT 1997. Lecture Notes in Computer Science*, vol 1408. Springer, Berlin, Heidelberg.
- [20] Colorni, Alberto, Marco Dorigo, and Vittorio Maniezzo. "Genetic algorithms and highly constrained problems: The time-table case." In *International conference on*

parallel problem solving from nature, Berlin, Heidelberg: Springer Berlin Heidelberg, 1990. 55-59.

- [21] Eiben, Ágoston Endre, Paul-Erik Raué, and Zs M. Ruttkay. Heuristic genetic algorithms for constrained problems. Vrije Universiteit, Department of Mathematics and Computer Science, Artificial Intelligence Group, 1993.
- [22] Mamede, Nuno, and Tiago Rente. "Repairing timetables using genetic algorithms and simulated annealing." In Proc. of PATAT, vol. 97, 187-204. 1997.
- [23] Corne, David, and John Ogden. "Evolutionary optimisation of methodist preaching timetables." In International Conference on the Practice and Theory of Automated Timetabling, Berlin, Heidelberg: Springer Berlin Heidelberg, 1997. 142-155.
- [24] Badoni, Rakesh P., Sanjeev Kumar, Mukesh Mann, R. P. Mohanty, and Ashirbad Sarangi. "Ant colony optimization algorithm for the university course timetabling problem using events based on groupings of students." In Modeling and Applications in Operations Research, CRC Press, 2024. 1-36.
- [25] Nwufoh, C. V., P. O. Achimugu, O. Achimugu, and T. D. Chollom. "A Hard Constraint Satisfaction Problem (Hcsp) Algorithm for University Course Time Tabling." 1st International Conference on Data Science and Engineering, Nigeria (6th-8th December, 2021) 203-211.