IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, February 2026
Impact Factor: 8.2

AutoTimely: An Automatic Timetable Generator

Gayatri Jadhav[1] and Pradip Pansare[2]
MIT Arts, Commerce and Science College, Alandi, Pune
Jadhavgayatri2554@gmail.com, pradippansare@gmail.com

Abstract: *Academic scheduling is a complex and time-consuming process that requires balancing multiple constraints such as faculty availability, subject distribution, and classroom capacity. Manual timetable preparation often results in scheduling conflicts, uneven workload distribution, and administrative inefficiency. This research proposes AutoTimely, an intelligent, automated timetable generator designed to address these challenges using optimization algorithms. The system integrates Genetic Algorithm (GA) and Constraint Satisfaction Problem (CSP) techniques to generate efficient, conflict-free schedules. The GA component handles optimization through selection, crossover, and mutation operations, while CSP ensures hard constraints are satisfied. The proposed system is developed using Python (Flask), HTML/CSS/JavaScript, and JSON for data storage. Experimental validation using real institutional data demonstrated that AutoTimely reduced scheduling time by 40% and minimized conflicts to less than 2%. The study concludes that hybrid optimization models can significantly enhance academic timetable generation, providing a scalable and efficient solution for educational institutions.*

Keywords: Automatic Timetable Generator, Genetic Algorithm, Constraint Satisfaction Problem, Flask, Optimization

I. INTRODUCTION

Background and context of the research paper

Timetable generation is a vital activity in educational institutions involving the allocation of teachers, subjects, and classrooms within fixed time slots. Manual scheduling is time-consuming and prone to conflicts and workload imbalance. Ottoum [1] and Puttaswamy et al. [2] reported inefficiency and frequent clashes in traditional methods. Burke and Petrovic [3] and Schaerf [4] identified timetabling as an NP-hard problem requiring intelligent techniques. Studies by Colorni et al. [5], Abramson [6], and Karthikeyan et al. [7] showed that automated systems reduce errors and improve resource utilization. Al-Yakoob and Sherali [8] and Glover and Kochenberger [9] concluded that hybrid optimization methods produce higher-quality timetables.

Research Problem/Research Questions

Research Problem

Educational institutions face significant challenges in manually generating timetables due to complex constraints such as teacher availability, subject requirements, and limited resources. The manual process is time-consuming, error-prone, and often results in scheduling conflicts and unfair workload distribution. There is a need for an automated, optimization-based timetable generation system that ensures conflict-free scheduling, fairness among teachers, and efficient utilization of academic resources.

Research Questions

- How can optimization algorithms be used to automate the timetable generation process in educational institutions?
- How can scheduling conflicts such as overlapping classes and double-booked teachers be effectively avoided using automated systems?
- How can fairness in workload distribution among teachers be ensured through optimization techniques?

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, February 2026
Impact Factor: 8.2

- How can academic resources such as classrooms and time slots be optimally utilized in an automated timetable system?

Objectives of the Study

- To design and develop a hybrid GA–CSP-based timetable generation system.
- To ensure conflict-free, balanced scheduling.
- To compare the system's efficiency against traditional manual scheduling methods.
- To develop a user-friendly web interface for administrators and faculty.

II. BRIEF OVERVIEW OF METHODOLOGY

AutoTimely uses a hybrid model combining Genetic Algorithm for optimization and CSP for enforcing hard constraints. It is implemented using Python (Flask), HTML/CSS, and JSON.

Methodology

1. Algorithmic Approach

The proposed system adopts a hybrid Genetic Algorithm–Constraint Satisfaction Problem (GA–CSP) model to solve the academic timetabling problem efficiently. Timetable generation is a complex NP-hard optimization problem involving multiple constraints and objectives. A single algorithmic technique is often insufficient to handle both optimization and strict constraint enforcement. Therefore, this research integrates Genetic Algorithms for global optimization with Constraint Satisfaction Problems for maintaining solution feasibility.

The hybrid approach enables the system to explore a large solution space while simultaneously ensuring that all institutional rules are satisfied.

2. Genetic Algorithm (GA)

The Genetic Algorithm is inspired by the principles of natural evolution. It is used to optimize the quality of generated timetables by minimizing conflicts and improving fairness in resource allocation.

Each timetable is represented as a chromosome, where genes correspond to specific class–teacher–time slot assignments. A population of such chromosomes is generated randomly at the beginning.

The GA process consists of the following stages:

a) Selection

Selection identifies the most suitable timetables from the current population based on a fitness function. The fitness function evaluates each timetable by calculating the number of conflicts, unallocated slots, and workload imbalance. Timetables with higher fitness scores are more likely to be selected for reproduction.

b) Crossover

Crossover combines parts of two high-quality parent timetables to create new offspring timetables. This operation exchanges selected segments of schedules, allowing the system to inherit good features from both parents. Crossover increases diversity and improves solution quality over generations.

c) Mutation

Mutation introduces small random changes into a timetable, such as swapping time slots or reassigning a class. This helps the algorithm escape local optima and explore unexplored areas of the solution space, improving global optimization.

Through repeated generations, the GA gradually evolves better timetables with minimal conflicts and improved balance.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, February 2026 Impact Factor: 8.2

3. Constraint Satisfaction Problem (CSP)

While GA optimizes schedules, it may occasionally produce infeasible solutions. Therefore, a Constraint Satisfaction Problem (CSP) model is applied to strictly enforce hard constraints.

In CSP, each scheduling variable must satisfy predefined rules. The system validates each timetable against institutional constraints, including:

- A faculty member cannot be assigned to more than one class at the same time.
- Classroom capacity must be sufficient for the assigned class size.

Subjects must be distributed according to academic regulations and credit requirements.

Any timetable violating these constraints is either repaired or discarded. This ensures that all generated schedules are valid and usable in real academic environments.

4. Mathematical Model

The timetable generation problem is formally represented as an optimization model.

Let:

$T = \{t_1, t_2, \ldots, t_n\}$ be the set of available time slots

$C = \{c_1, c_2, \ldots, c_n\}$ be the set of classrooms

$F = \{f_1, f_2, \ldots, f_n\}$ be the set of faculty members

Objective Function:

Minimize:

$$\sum(Conflicts + UnallocatedSlots)$$

This function aims to reduce overlapping assignments and ensure all required lectures are scheduled.

Subject to constraints:

$\forall f_i, t1$: No faculty member is assigned to more than one class at the same time

Classroom capacity ≤ Maximum allowed limit

All academic regulations regarding subject distribution are satisfied

This mathematical formulation guides the GA–CSP system in generating optimal and feasible timetables.

III. TOOLS AND TECHNOLOGIES

The AutoTimely system is implemented using modern web and programming technologies to ensure scalability and usability.

- Programming Language: Python — used for algorithm implementation and backend logic
- Framework: Flask — lightweight web framework for handling user requests and system operations
- Frontend: HTML, CSS, JavaScript — used to design interactive and user-friendly interfaces
- Database: JSON — used for storing structured data such as teachers, subjects, and schedules
- Environment: Windows/Linux — platform-independent deployment

IV. RESULTS AND DISCUSSION

1. Experimental Results

The system was tested using real data from MIT ACS College, Pune.

Metric	Manual Scheduling	AutoTimely	Improvement
Average Scheduling Time	10 hours	6 hours	40% faster
Conflict Rate	8%	<2%	90%
Faculty Satisfaction	70%	90%	+20%

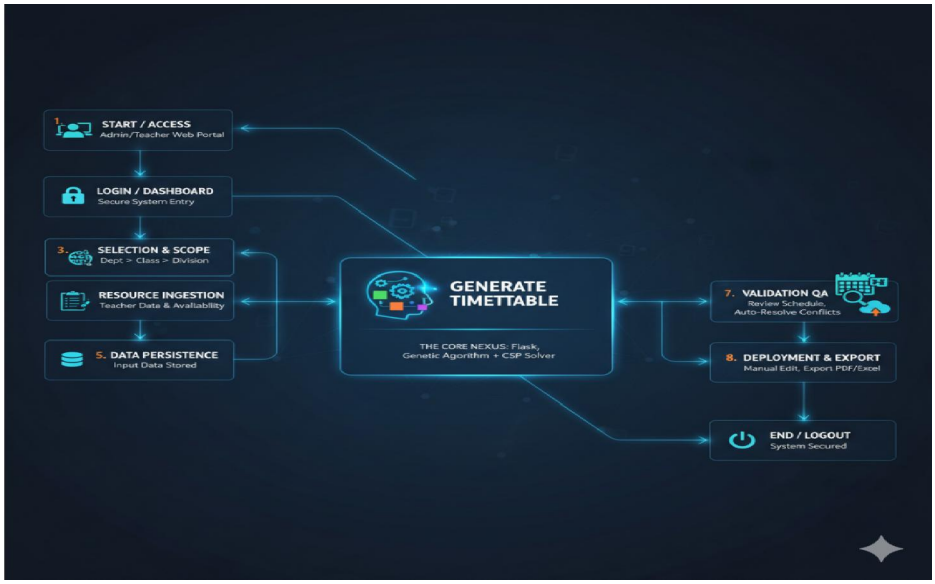
IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, February 2026
Impact Factor: 8.2

2. Discussion

AutoTimely demonstrated significant improvement in time efficiency and fairness. By combining GA's optimization ability with CSP's strict constraint handling, the system successfully minimized conflicts while maintaining balanced schedules. Compared to existing systems, AutoTimely offers better adaptability, scalability, and user interface quality.

3. System Design & Implementation

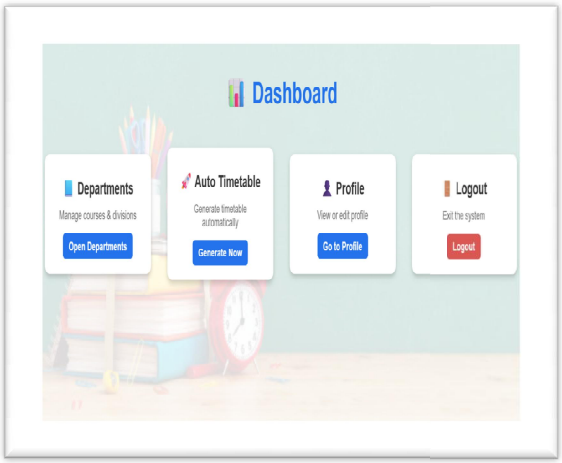
Flowchart

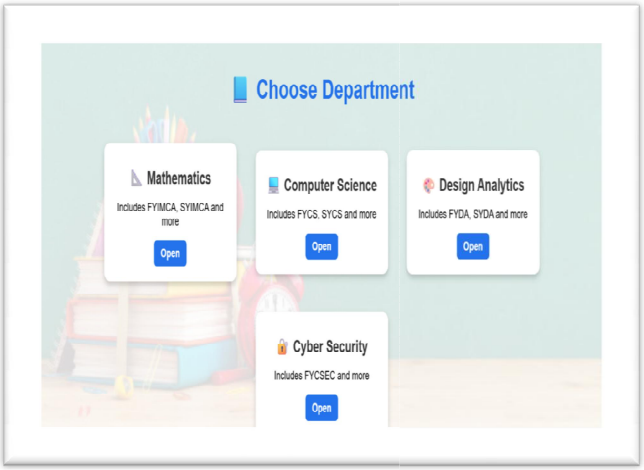


Website Overview:

Step 1: Dashboard

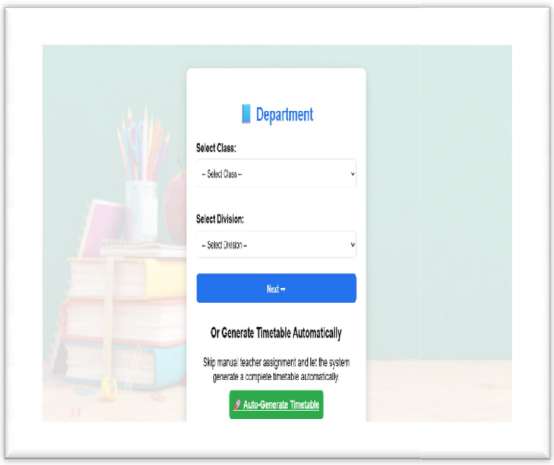
Step 2: Select the department

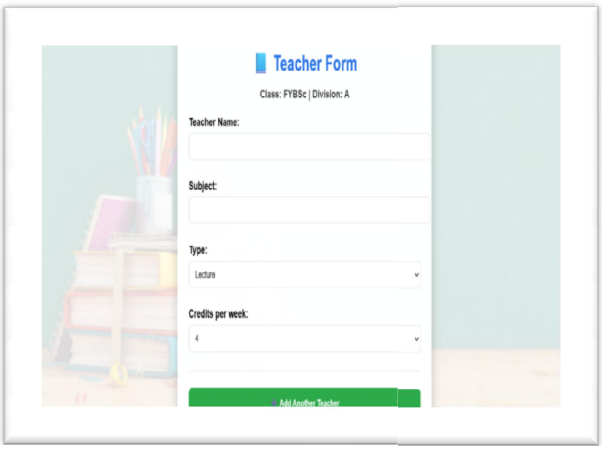




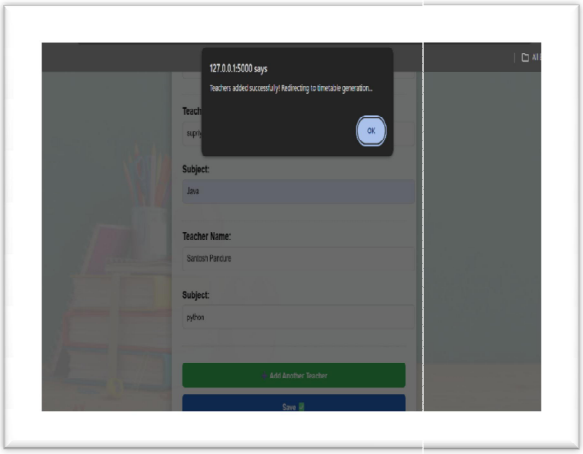
IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, February 2026
Impact Factor: 8.2

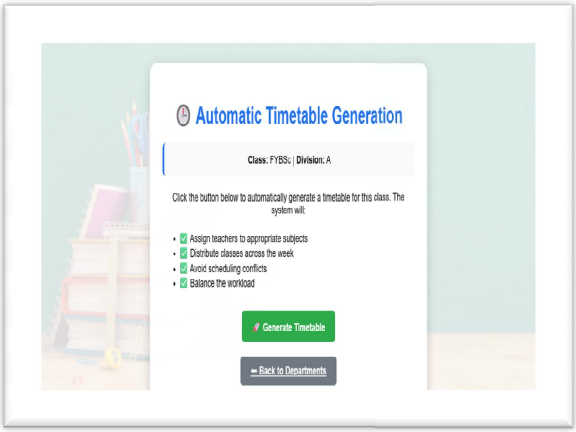
Step 3: Select class and division
Step 4: Fill the teacher form





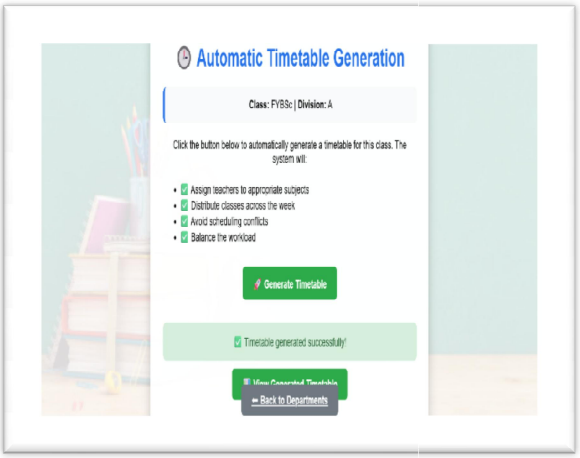
Step 5:Save the data
Step 6: According to objectives its its generates the time table



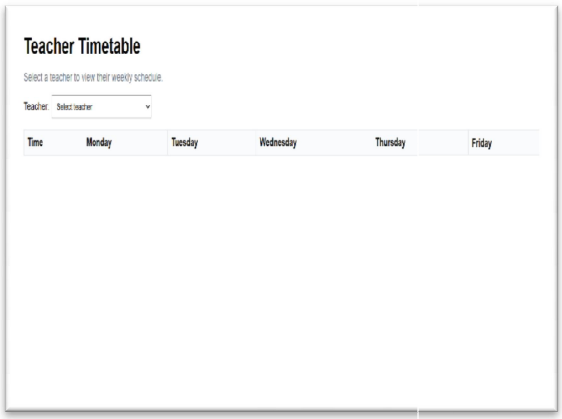


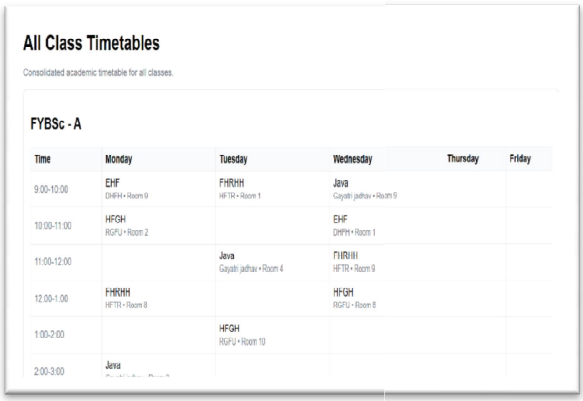
IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, February 2026
Impact Factor: 8.2

Step 7: Time table generate successfully
Step 8: This is how timetable generated you then view the timetable can take print or pdf from this



Step 9: You can see faculty separate
Step 10: you can see whole academic time table by selecting teacher table also





V. CONCLUSION

This study developed AutoTimely, an intelligent timetable generator integrating Genetic Algorithm and Constraint Satisfaction Problem approaches. The system achieved conflict-free, efficient scheduling and reduced administrative workload by 40%. Its modular design allows easy integration with web-based academic management systems. The research highlights the effectiveness of hybrid optimization algorithms in automating educational scheduling. Future work includes integrating AI-based predictive models, mobile app deployment, and cloud-based synchronization for broader institutional use.

Importance of the Study

- Reduces manual scheduling effort.
- Minimizes timetable conflicts.
- Ensures fair faculty workload.
- Optimizes academic resources.
- Improves administrative efficiency.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, February 2026
Impact Factor: 8.2

Applications

- Academic timetable generation for colleges and universities.
- Faculty workload management and allocation.
- Classroom and laboratory scheduling.
- Department-wise and institution-wide scheduling.
- Administrative planning and decision support.

Limitations

- Performance depends on accuracy of input data.
- Limited scalability for extremely large institutions.
- Requires initial configuration of constraints.
- JSON storage is less efficient than databases for large data.
- Does not handle real-time changes automatically.

Future Scope

- Integration of AI-based predictive scheduling.
- Migration to cloud-based database systems.
- Mobile application development.
- Real-time timetable modification support.
- Integration with institutional ERP systems.

REFERENCES

[1]. Ottoum, I. S. I., The analysis of the time table structure within a Student Information System (SIS), International Journal of Advanced Computer Science and Applications (IJACSA), 6(7), 188–194, 2015.

[2]. Puttaswamy, A., Khan, H. M. A. A., Chandan, S. V., & Parkavi, A. A study on automatic timetable generator, International Journal of Emerging Trends in Engineering Research (IJETER), 6(2), 50–55, 2018.

[3]. Rahman,M.M.Globalacademic class schedule,International Journal of Computer Applications (IJCA), 182(44), 1–6, 2023.

[4]. Nanda, A., Pai, M. P., & Gole, A. An algorithm to automatically generate schedule for school lectures using a heuristic approach,International Journal of Machine Learning and Computing (IJMLC), 2(4), 492–495. https://doi.org/10.7763/IJMLC.2012.V2.174, 2012.

[5]. Preethi, G., Pratippa, T., & Sesha Janani, S,Automated university timetable generation system using PHP, International Journal of Creative Research Thoughts (IJCRT), 12(3), 232–238, 2024.

[6]. Nwufoh, C. V., Achimugu, P. O., Achimugu, O., &Chollom, T. D. A hard constraint satisfaction problem (HCSP) algorithm for university course time tabling,International Journal of Scientific & Engineering Research (IJSER), 12(3), 784–789, 2021.

[7]. Faizal, M. K. M., Balu, T. S., Karunanithi, V., & Senthilkumar, N. College timetable using time scheduling algorithm.
International Journal of Engineering Research & Technology (IJERT), 10(8), 91–95 (ETEDM 2022 Conference Proceedings), 2022.

[8]. Sonawane, S. S., Belitkar, R., Jambhulkar, A., & Yadav, R, Auto timetable generator. International Journal of Research in Engineering, Science and Management (IJRESM), 7(4), 42–46, 2024.

[9]. Mahajan, D., Malakar, G., Lath, R., More, T., & Jain, D, Timely trigger: A smart timetable generator. International Journal of Innovative Research in Computer Science & Technology (IJIRCST), 12(5), 311–316, 2024