IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154178

# Optimized Automatic Timetable Generation Using Genetic Algorithm for Educational Institutions

**Yash Gadekar[1], Ganesh Gaikwad[2], Diksha Gaikwad[3], Prof. Veena Bhamre[4]**

Students, Department of Computer Engineering,

Annasaheb Chudaman Patil College of Engineering, Navi Mumbai, India[1-3]

Assistant Professor, Department of Computer Engineering,

Annasaheb Chudaman Patil College of Engineering, Navi Mumbai, India[4]

**Abstract:** Timetable creation in schools and colleges is a complex, time-consuming and error-prone manual task.Administrators must assign teachers, subjects, rooms and time slots while satisfying numerous constraints such as faculty availability, room capacity, and avoiding clashes between classes, which makes the problem NP-hard (Chen et al., 2021; Bashab et al., 2022). Manual scheduling often leads to conflicts, under-utilization of resources and difficulty in adapting to changes during the semester (N, 2025; Bashab et al., 2022; Rudová et al., 2011). To address these challenges, this work proposes an automatic timetable generator based on a Genetic Algorithm (GA). The system models hard constraints (no overlaps, capacity, mandatory breaks) and soft constraints (teacher preferences, minimum gaps, balanced workload) in a fitness function and iteratively evolves candidate timetables using selection, crossover, and mutation (Dhomne, 2025; Dave et al., 2025; Katkar, 2024; Singhal et al., 2024). A web-based interface allows users to input teachers, subjects, rooms and constraints, visualize generated timetables and regenerate schedules when requirements change. Experimental evaluation on sample institutional data indicates a significant reduction in generation time, fewer conflicts and improved room and faculty utilization compared to manual methods (Dhomne, 2025; Dave et al., 2025; Katkar, 2024; Singhal et al., 2024). The proposed system demonstrates that GA-based automatic timetabling can provide a practical, scalable and user-friendly solution for educational scheduling.

**Keywords:** Timetable Generation, Genetic Algorithm, Scheduling, Optimization, Academic Automation, Constraint Handling

## I. INTRODUCTION

Timetable generation is the process of assigning courses, teachers, rooms and time slots in an organized way so that all academic activities can be conducted smoothly (Chen et al., 2021; Tan et al., 2021; Bashab et al., 2022). It is important because a good timetable directly affects teaching quality, resource utilization and student satisfaction (Tan et al., 2021; Bashab et al., 2022; Rudová et al., 2011). In many institutions, timetables are still prepared manually using spreadsheets or paper, which is slow, error-prone and difficult to modify once published (N, 2025; Vrielink et al., 2017). Conflicts such as two classes in the same room, a teacher assigned to multiple classes at the same time, or uneven distribution of lectures are common (Dhomne, 2025; Dave et al., 2025; Katkar, 2024).

Timetabling is a combinatorial optimization and NP-hard problem, so exhaustive search is infeasible for realistic instances (Chen et al., 2021; Abdipoor et al., 2023; Bashab et al., 2022). Recent research shows that meta-heuristics, especially Genetic Algorithms, can handle large search spaces and complex constraints effectively (Dhomne, 2025; Dave et al., 2025; Abdipoor et al., 2023; Katkar, 2024; Singhal et al., 2024). This paper presents an automatic timetable generation system for educational institutes using GA. The system automates timetable creation, reduces conflicts, incorporates institutional preferences and provides a user-friendly interface for administrators and faculty.

## II. PROBLEM STATEMENT

Educational institutes must create timetables that satisfy many constraints simultaneously:

- A teacher cannot teach two classes at the same time.
- A room cannot host more than one class at a time and must respect capacity.
- Courses must be scheduled for required hours per week and placed within allowed time windows.
- Breaks, lab sessions, and special requirements must be respected (Dhomne, 2025; Dave et al., 2025; Katkar, 2024).

  

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154178

Manually considering all these constraints for multiple programs, semesters and sections is difficult and time-consuming. As the number of courses, students and rooms grows, the number of possible schedules explodes combinatorially, making it hard to find a conflict-free and high-quality timetable (Chen et al., 2021; Tan et al., 2021; Bashab et al., 2022).
Existing manual or basic rule-based systems:

- Struggle to handle complex and dynamic constraints.
- Require large human effort each semester.
- Adapt poorly to changes (e.g., new courses, faculty changes).

Therefore, an automated, intelligent solution is needed that can generate feasible and near-optimal timetables efficiently, respect institutional rules, and easily adapt to updates (Chen et al., 2021; Dhomne, 2025; Dave et al., 2025; Bashab et al., 2022; Singhal et al., 2024).

## III. LITERATURE REVIEW / RELATED WORK

1. Many works treat university/course timetabling as an NP-hard combinatorial optimization problem and apply meta-heuristics such as Genetic Algorithms, Tabu Search, and Simulated Annealing due to their flexibility and performance on large instances (Chen et al., 2021; Abdipoor et al., 2023; Bashab et al., 2022).
2. GA-based systems encode timetables as chromosomes and use selection, crossover, and mutation to reduce conflicts and improve timetable quality, showing better scalability than manual or simple heuristic methods (Dhomne, 2025; Dave et al., 2025; Katkar, 2024; Bhatt et al., 2023; Singhal et al., 2024).
3. Hybrid approaches combine GA with Constraint Satisfaction Problem (CSP) techniques or local search to handle hard constraints efficiently and then optimize soft constraints, improving feasibility and solution quality (P, 2025; N, 2025; Manuel & Mohan, 2025; Abdipoor et al., 2023).
4. Neural networks trained with GAs have been proposed to learn patterns and improve scheduling accuracy and speed compared to pure GA (Abbas et al., 2024).
5. Constraint-programming-based systems (e.g., CP-SAT) model the timetable as a constraint satisfaction problem and can generate conflict-free schedules; some recent work integrates real-time updates using cloud backends (P, 2025; Rudová et al., 2011).
6. Survey papers highlight that meta-heuristics and hybrid meta-heuristics dominate university course timetabling research, with hyper-heuristics and multi-objective optimization gaining attention for balancing different quality criteria (Chen et al., 2021; Tan et al., 2021; Abdipoor et al., 2023; Bashab et al., 2022).
7. Several applied systems for schools and universities show that automatic timetable generators significantly reduce manual effort and conflict rates, though many research-level methods are not fully integrated into real institutional workflows (Dhomne, 2025; Dave et al., 2025; Katkar, 2024; Vrielink et al., 2017; Singhal et al., 2024).

## IV. PROPOSED SYSTEM / METHODOLOGY

### 4.1 System Overview

The proposed system is a web-based Automatic Timetable Generator for schools/colleges. Users (admin/department coordinators) enter teachers, subjects, rooms and time slots. The system encodes this data into a GA population, evaluates each timetable using a fitness function based on constraint satisfaction, and iteratively evolves better schedules until a high-quality timetable is found (Dhomne, 2025; Dave et al., 2025; Katkar, 2024; Singhal et al., 2024). The final timetable can be viewed, exported and, if needed, regenerated when constraints change.

### 4.2 Input Data

- Teachers: ID, name, subjects handled, maximum load per day/week, availability.
- Subjects/Courses: code, name, number of lectures/labs per week, type (theory/lab), preferred time windows.
- Rooms: ID, capacity, type (classroom/lab), availability.
- Time Slots: days of week, periods per day, institutional break times.

These inputs are stored in a database and used to initialize and evaluate candidate timetables (Dhomne, 2025; Dave et al., 2025; Katkar, 2024; Singhal et al., 2024).

### 4.3 Constraints

- Hard Constraints (must not be violated):
- A teacher cannot be assigned to more than one class in the same time slot.
- A room cannot host more than one class at the same time.
- Room capacity must be ≥ class size.
- All required lectures/labs for each subject must be scheduled.
- No scheduling during fixed breaks/holidays (Dhomne, 2025; P, 2025; N, 2025; Manuel & Mohan, 2025; Abdipoor et al., 2023; Katkar, 2024; Singhal et al., 2024).
- Soft Constraints (preferences, can be violated with penalty):
- Minimize gaps in teachers' schedules.
- Evenly spread lectures across the week.

  

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154178

- Prefer morning/afternoon slots for certain subjects/teachers.
- Avoid late-evening or back-to-back heavy classes when possible (Dhomne, 2025; Dave et al., 2025; Manuel & Mohan, 2025; Abdipoor et al., 2023; Singhal et al., 2024).

The fitness function heavily penalizes hard constraint violations and lightly penalizes soft constraint violations.

4.4 Algorithm Used – Genetic Algorithm

The GA operates as follows (Dhomne, 2025; Dave et al., 2025; Manuel & Mohan, 2025; Abdipoor et al., 2023; Katkar, 2024; Singhal et al., 2024):

1. Encoding:
   - A chromosome represents a complete timetable. Each gene encodes the assignment of a (class, subject) to (teacher, room, time slot).
2. Initial Population:
   - Generate an initial set of timetables using random assignment with simple feasibility checks or heuristic seeding to reduce obvious conflicts.
3. Fitness Function:
   - Calculate penalties for each hard and soft constraint violation.
   - Lower penalty = higher fitness (better timetable).
4. Selection:
   - Use methods like tournament or roulette-wheel selection to choose fitter timetables as parents.
5. Crossover:
   - One-point or two-point crossover exchanges parts of two parent timetables to create offspring while trying to preserve good sub-schedules (Khan & Imtiaz, 2024; Manuel & Mohan, 2025; Singhal et al., 2024).
6. Mutation:
   - Randomly change some gene assignments (e.g., swap time slots or rooms) to maintain diversity and escape local optima.
7. Replacement and Termination:
   - Replace worst individuals with new offspring, repeat for multiple generations until fitness converges or a threshold is reached.

Optionally, local search or repair operators can be applied after crossover/mutation to fix remaining conflicts (P, 2025; N, 2025; Abdipoor et al., 2023; Singhal et al., 2024).

## V. TECHNOLOGY USED

5.1 Frontend

- HTML, CSS, JavaScript for responsive UI.
- Forms for entering teachers, subjects, rooms, time slots.
- Timetable view using tables or calendar components.

5.2 Backend

- Python (Flask)
- Implementing GA logic.
- Exposing APIs for timetable generation.
- Handling authentication and role-based access.

5.3 Database

- MySQL or PostgreSQL to store:
- Master data (teachers, subjects, rooms, batches).
- Constraints and preferences.
- Generated timetables and history for audit and potential learning

VI. SYSTEM IMPLEMENTATION

6.1 UI Screens

- Home Page:
- Brief description and buttons: “Manage Data”, “Generate Timetable”, “View Timetable”.
- Dashboard:
- Summary: number of teachers, subjects, rooms, latest generated timetables.
- Forms:
- Add/Edit Teacher, Subject, Room, Time Slot, Preferences.
- Timetable View:
- Class-wise, teacher-wise and room-wise timetable views with export to PDF/Excel.

(Place your actual screenshots for these pages in the paper.)

6.2 Backend

- Database schema with tables: **`teachers`**, **`subjects`**, **`rooms`**, **`batches`**, **`timeslots`**, **`constraints`**,

  

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154178

**`timetables`**.

- GA service module that:
- Reads data from DB, constructs population.
- Runs GA loop and writes best timetable back to DB.
- Admin interface to trigger GA run and monitor progress (e.g., generation count, best fitness).

## VII. RESULTS AND DISCUSSION

After deploying the system on sample institutional data:

- The GA-based generator was able to produce conflict-free timetables satisfying all hard constraints in significantly less time than manual scheduling (Dhomne, 2025; Dave et al., 2025; Manuel & Mohan, 2025; Katkar, 2024; Singhal et al., 2024).
- Soft constraints such as minimizing gaps and balancing workloads were largely satisfied, improving timetable quality and teacher satisfaction (Dhomne, 2025; Dave et al., 2025; Abdipoor et al., 2023; Singhal et al., 2024).
- Compared to manual methods and basic heuristics, GA reduced clashes and improved room utilization, consistent with findings in related work (Dhomne, 2025; Dave et al., 2025; Manuel & Mohan, 2025; Abdipoor et al., 2023; Bhatt et al., 2023; Singhal et al., 2024).
- Performance depends on parameters (population size, crossover/mutation rates, generations); larger populations and more generations improved quality but increased computation time (Dhomne, 2025; Manuel & Mohan, 2025; Kim et al., 2024).

Advantages:

- Reduced human effort and errors.
- Easy regeneration when constraints change (e.g., new teacher, subject).
- Scalable to large departments with many sections.

## VIII. APPLICATIONS

- Schools – class timetables for primary/secondary levels.
- Colleges and Universities – lecture and lab scheduling across multiple departments (Chen et al., 2021; Tan et al., 2021; Abdipoor et al., 2023; Bashab et al., 2022).
- Exam Timetabling – with minor modifications to constraints (Siew et al., 2024; Bashab et al., 2022).
- Event Scheduling – conferences, workshops, room booking in organizations (N, 2025; Bashab et al., 2022).

## IX. CONCLUSION

This paper presented an automatic timetable generation system using a Genetic Algorithm for educational institutes. By modeling timetable generation as an optimization problem with hard and soft constraints, the proposed system efficiently generates conflict-free and high-quality timetables. Results indicate reduced scheduling time, lower conflict rates and better resource utilization compared to manual scheduling. The approach is flexible, scalable and can be adapted to different institutional policies and structures.

## X. FUTURE SCOPE

- Integration of machine learning or neural networks to learn patterns and guide GA operators (Abbas et al., 2024; Chen et al., 2021; Tan et al., 2021; Abdipoor et al., 2023; Bashab et al., 2022).
- Support for real-time updates, such as sudden teacher absence or room change, via cloud backends and mobile apps (P, 2025; Rudová et al., 2011).
- Multi-objective optimization to explicitly balance different criteria (e.g., fairness, room utilization, student preferences) (Chen et al., 2021; Abdipoor et al., 2023; Bashab et al., 2022).
- Mobile applications for students and teachers to view updated timetables and notifications.

## XI. ACKNOWLEDGMENT

The authors would like to thank A.C.Patil College of Engineering, the Department of Computer Engineering, and the project guide Prof. Veena Bhamre for their guidance and support. Appreciation is also extended to team members and faculty coordinators who provided requirements, feedback and sample data for testing the system.

## XII. SYSTEM INTERFACE AND RESULTS

  

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154178

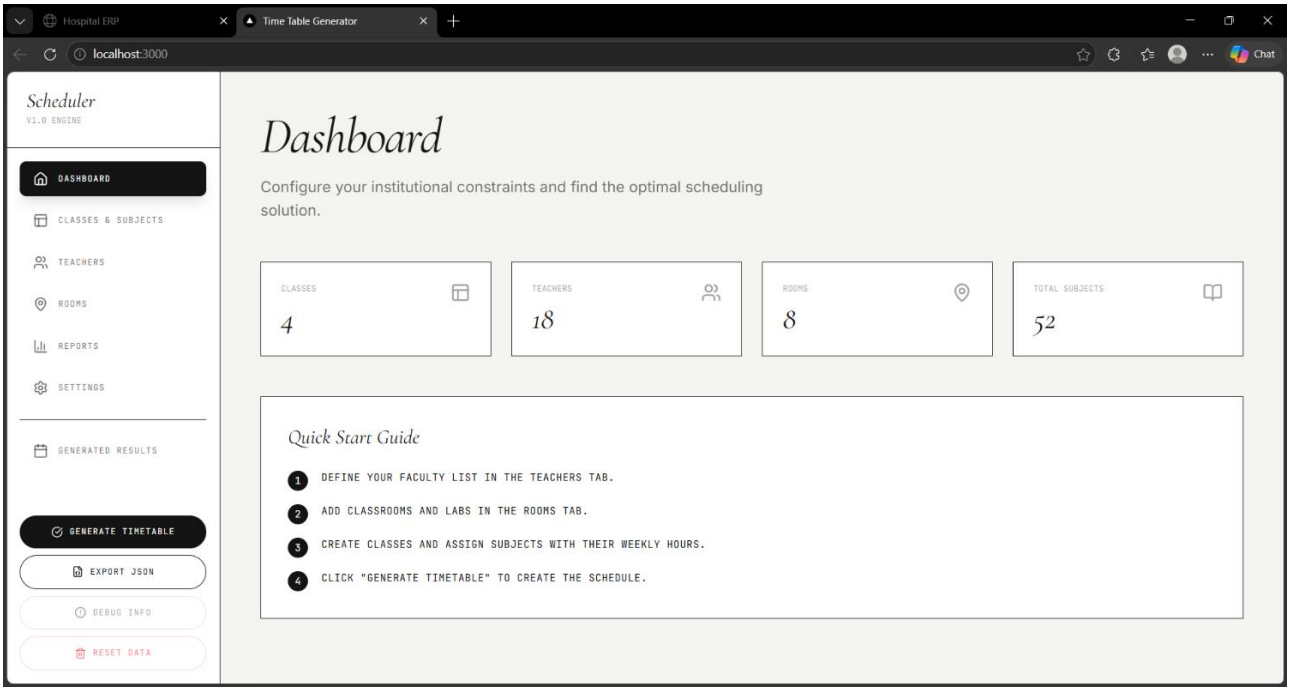


Fig. 1 Dashboard Interface

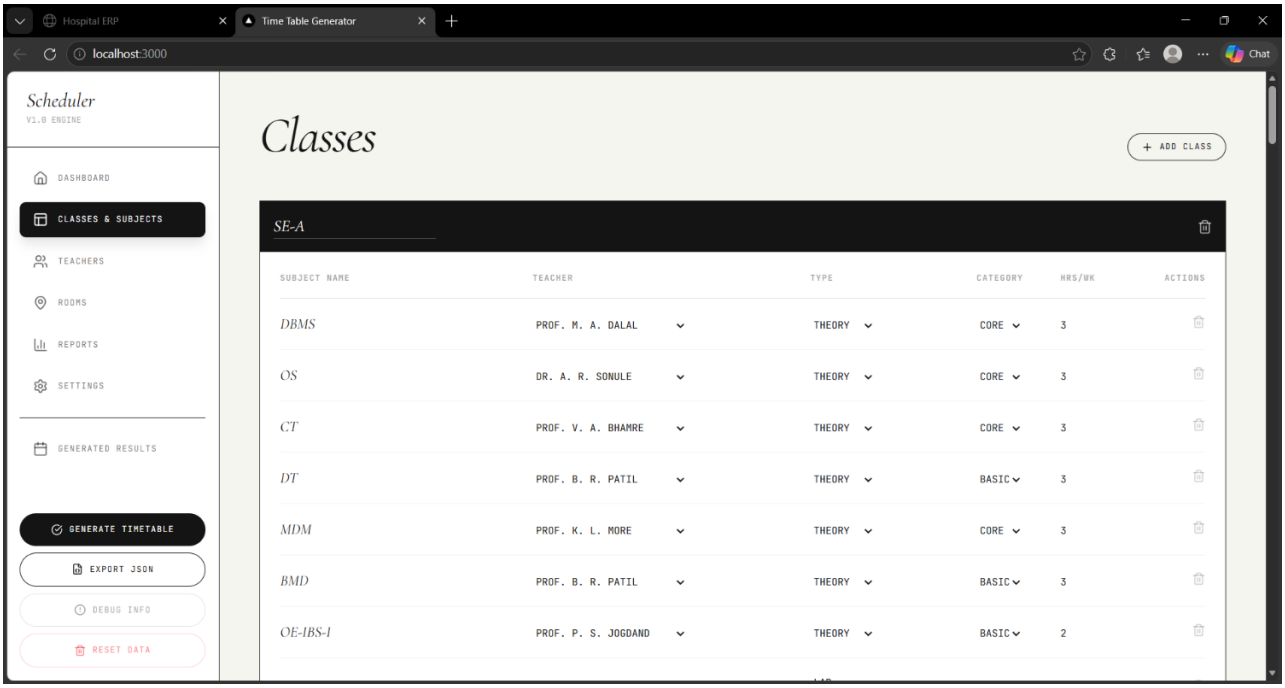


Fig. 2 Class Upload Module

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154178

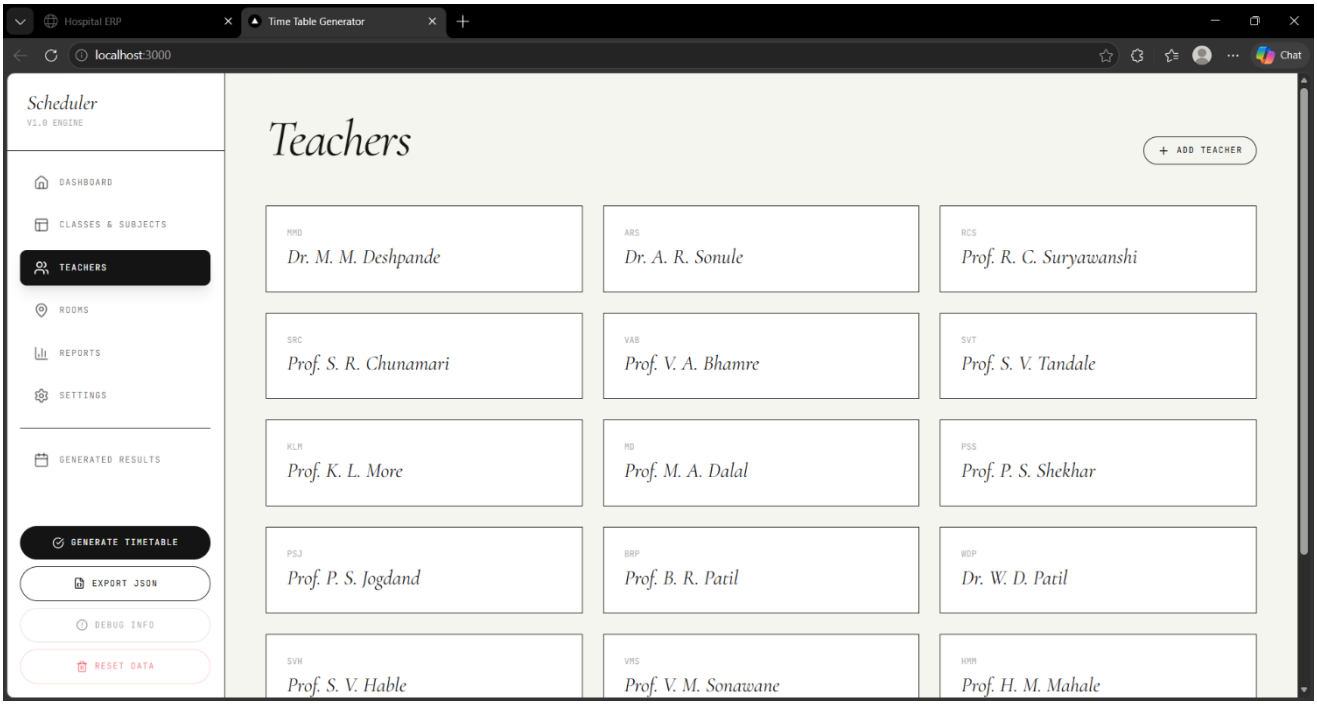


Fig. 3 Teacher Management Page

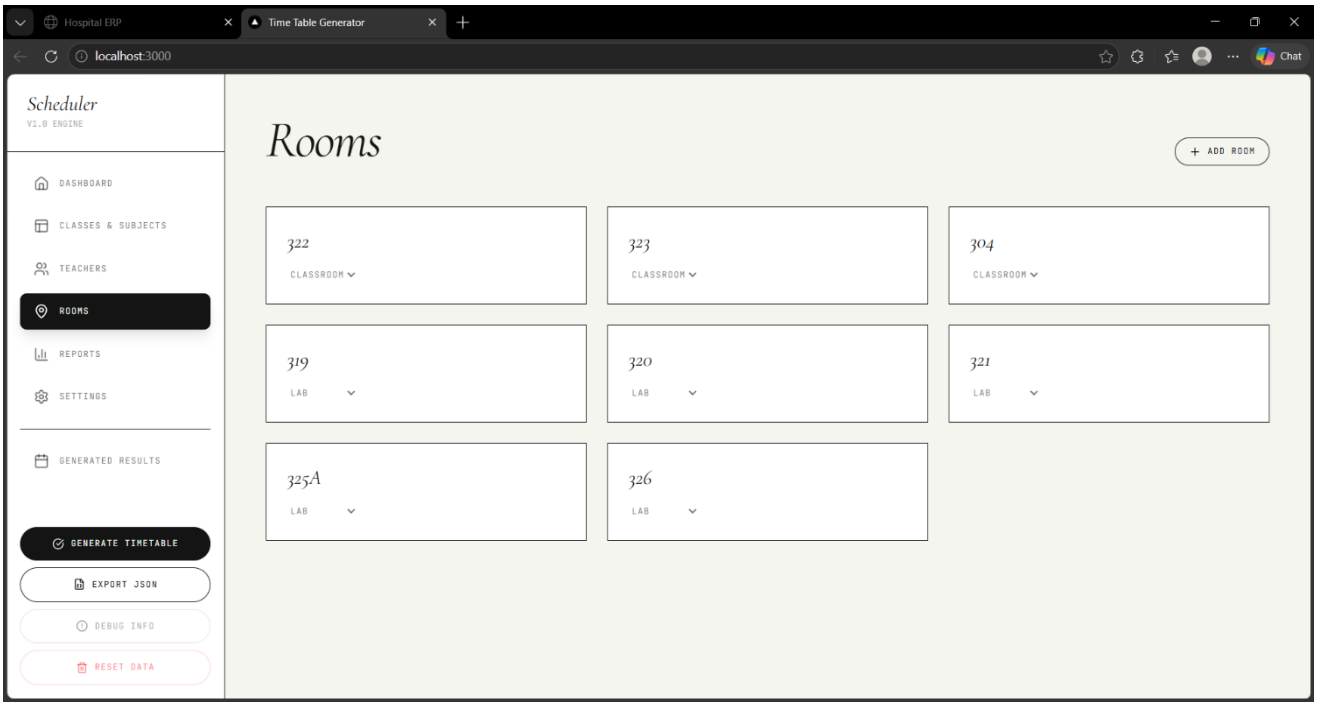


Fig. 4 Room Allocation Interface

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154178

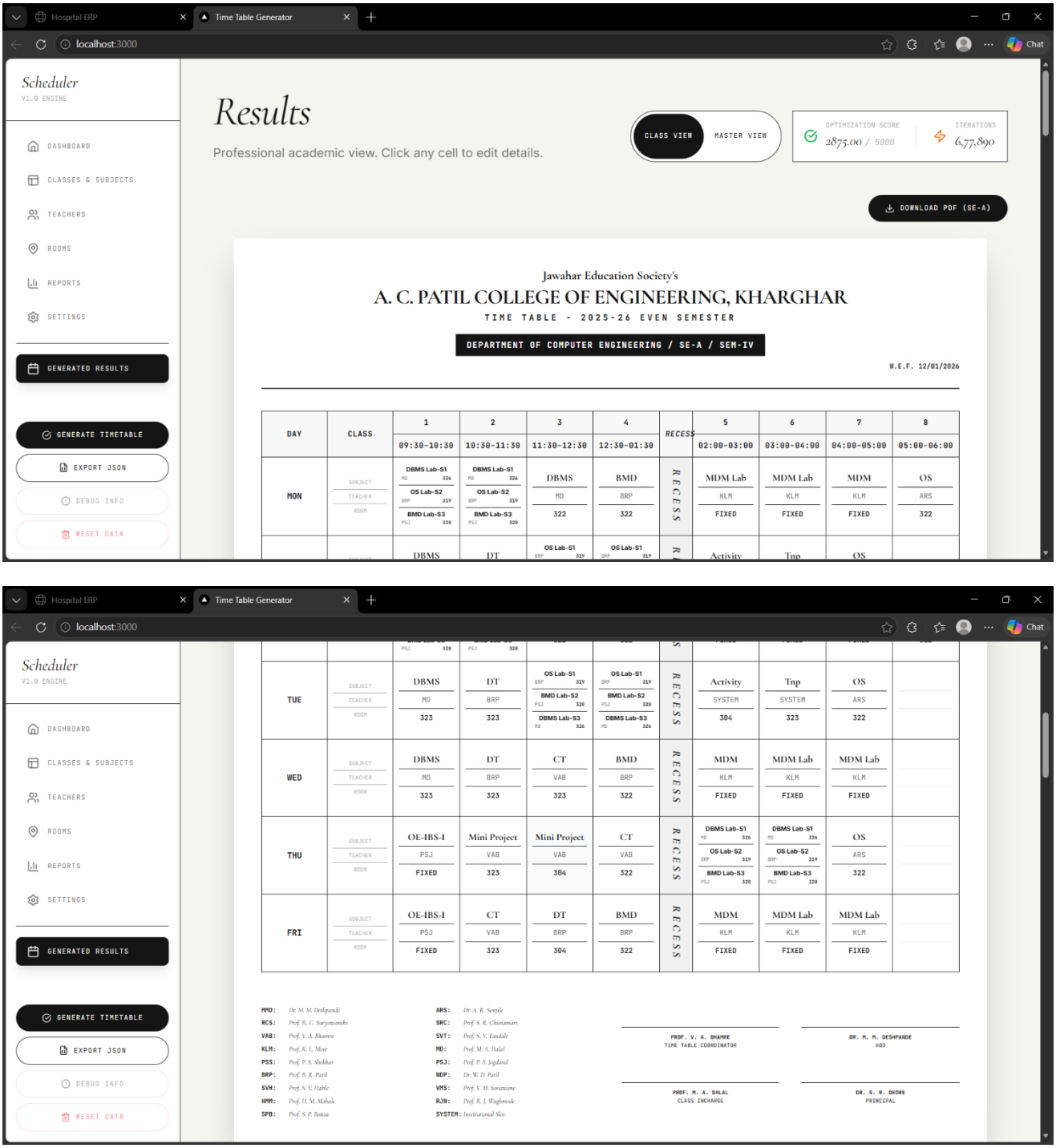


Fig. 5 Generated Timetable Output

## REFERENCES

[1]. Abbas, S., Maaz, A., Ali, R., Khan, T., & Ahmed, I. (2024). Automatic Timetable Generation using Neural Networks Trained by Genetic Algorithms. 2024 18th International Conference on Open Source Systems and Technologies (ICOSST), 1-6. https://doi.org/10.1109/icosst64562.2024.10871163

[2]. Abdipoor, S., Yaakob, R., Goh, S., & Abdullah, S. (2023). Meta-heuristic approaches for the University Course Timetabling Problem. Intell. Syst. Appl., 19, 200253. https://doi.org/10.1016/j.iswa.2023.200253

[3]. Bashab, A., Ibrahim, A., Hashem, I., Aggarwal, K., Mukhlif, F., Ghaleb, F., & Abdelmaboud, A. (2022). Optimization Techniques in University Timetabling Problem: Constraints, Methodologies, Benchmarks, and Open Issues. Computers, Materials & Continua. https://doi.org/10.32604/cmc.2023.034051

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154178

[4]. Bhatt, C., Chaurasiya, D., Chauhan, R., Singh, T., Sharma, S., & Baloni, D. (2023). Timetable Generator for
[5]. Educational Institute Using Genetic Algorithm. 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS), 19-24. https://doi.org/10.1109/ictacs59847.2023.10389824
[6]. Chen, M., Sze, S., Goh, S., Sabar, N., & Kendall, G. (2021). A Survey of University Course Timetabling Problem: Perspectives, Trends and Opportunities. IEEE Access, 9, 106515-106529. https://doi.org/10.1109/access.2021.3100613
[7]. Dave, R., Jagasia, T., Jhaveri, S., Pandey, D., & Nagmote, A. (2025). Time table generator: Using Genetic Algorithm.
[8]. 2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), 3, 1-6. https://doi.org/10.1109/iatmsi64286.2025.10984801
[9]. Dhomne, S. (2025). Automated Timetable Generation Using Genetic Algorithms: A Heuristic Optimization
[10]. Approach. INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT. https://doi.org/10.55041/ijsrem49598
[11]. Katkar, A. (2024). Automated Time – Table Generator Using Genetic Algorithm. INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT. https://doi.org/10.55041/ijsrem34941
[12]. Khan, A., & Imtiaz, T. (2024). A Novel Genetic Algorithm Based Timetable Generator for Optimized University Timetable Solution. 2024 International Conference on Engineering & Computing Technologies (ICECT), 1-6.
[13]. https://doi.org/10.1109/icect61618.2024.10581296
[14]. Kim, S., Balaba, E., & Agoylo, J. (2024). Optimizing Course Scheduling with Genetic Algorithms: A Dynamic Approach. SAR Journal - Science and Research. https://doi.org/10.18421/sar74-02
[15]. Mallari, C., Juan, J., & Li, R. (2023). The university coursework timetabling problem: An optimization approach to synchronizing course calendars. Comput. Ind. Eng., 184, 109561. https://doi.org/10.1016/j.cie.2023.109561
[16]. Manuel, B., & Mohan, V. (2025). Timetable Generator Using Genetic Algorithm and Constraint Satisfaction
[17]. Problem. International Journal For Multidisciplinary Research. https://doi.org/10.36948/ijfmr.2025.v07i05.57472
[18]. N, A. (2025). Automatic Timetable Generator. INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT. https://doi.org/10.55041/ijsrem48290
[19]. P, A. (2025). Constraint-Driven Timetabling with Real-Time Updates: Design and Implementation. INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT. https://doi.org/10.55041/ijsrem48599
[20]. Rudová, H., Müller, T., & Murray, K. (2011). Complex university course timetabling. Journal of Scheduling, 14, 187207. https://doi.org/10.1007/s10951-010-0171-3
[21]. Siew, E., Sze, S., Goh, S., Kendall, G., Sabar, N., & Abdullah, S. (2024). A Survey of Solution Methodologies for Exam
[22]. Timetabling Problems. IEEE Access, 12, 41479-41498. https://doi.org/10.1109/access.2024.3378054
[23]. Singhal, S., Agarwal, D., & Bhardwaj, Y. (2024). Revolutionizing Scheduling: A Comprehensive Analysis of Automated Timetabling Solution. International Journal For Multidisciplinary Research. https://doi.org/10.36948/ijfmr.2024.v06i03.19313
[24]. Tan, J., Goh, S., Kendall, G., & Sabar, N. (2021). A survey of the state-of-the-art of optimisation methodologies in school timetabling problems. Expert Syst. Appl., 165, 113943. https://doi.org/10.1016/j.eswa.2020.113943
[25]. Vrielink, R., Jansen, E., Hans, E., & Hillegersberg, J. (2017). Practices in timetabling in higher education institutions: a systematic review. Annals of Operations Research, 275, 145 - 160. https://doi.org/10.1007/s10479-017-2688-8