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 10, April 2026
Impact Factor: 8.2

Time Table AI : Automated Academic Timetable Generation System

Vignesh Totawar[1], Suchit Khobragade[2], Sandesh Bansod[3], Ramkishan Jadhav[4], Dr. M. D. Tambakhe[5]
Students, Department of Information Technology[1-4]
Professor, Department of Information Technology[5]
SIPNA College of Engineering and Technology, Amravati, Maharashtra, India

Abstract: *Time Table AI is an automated academic scheduling system designed to replace the complex and error-prone manual timetable creation process. It uses a modern tech stack with React, TypeScript, Node.js, Express, and Supabase PostgreSQL. The core system is based on an AI- driven Constraint Satisfaction Problem (CSP) model using greedy and backtracking algorithms. It ensures conflict-free schedules while optimizing faculty workload and resource utilization. The platform provides role-based dashboards for administrators, faculty, and students, improving efficiency and reducing administrative effort..*

Keywords: *Time Table AI*

I. INTRODUCTION

The creation and management of academic timetables is a critical yet complex task in educational institutions. A timetable acts as a structural blueprint that coordinates students, faculty, and resources. With increasing course offerings and student intake, scheduling complexity has grown significantly. Traditional manual methods are time-consuming and prone to errors. Administrators often struggle to balance constraints like faculty availability, room capacity, and course requirements. This leads to conflicts such as double bookings and inefficient resource utilization. Manual scheduling also fails to optimize soft constraints like workload distribution. These challenges highlight the need for an automated solution.

Time Table AI is introduced as an intelligent system to address these issues. It automates timetable generation using advanced algorithms. The system is based on the concept of Constraint Satisfaction Problems (CSP). It uses a hybrid approach combining greedy algorithms and backtracking. This ensures conflict-free schedules while optimizing resource usage. The platform is built as a full-stack web application using modern technologies. It provides role-based access for administrators, faculty, and students. Overall, it reduces human effort, minimizes errors, and improves efficiency in academic scheduling.

II. LITERATURE REVIEW

The University Timetabling Problem (UTP) is a well-known challenge in Operations Research and Computer Science. It is classified as an NP-complete problem due to its high complexity. The task involves assigning lectures, labs, and seminars to time slots and resources. This must be done while satisfying multiple constraints. Over time, research has shifted from manual methods to algorithmic solutions.

Initially, timetabling was handled manually by administrators. This process relied heavily on human experience and intuition. However, it often resulted in errors such as scheduling conflicts. Double-booking of rooms and overlapping faculty schedules were common issues. As institutions expanded, manual methods became inefficient and impractical. This led to the adoption of computational approaches.

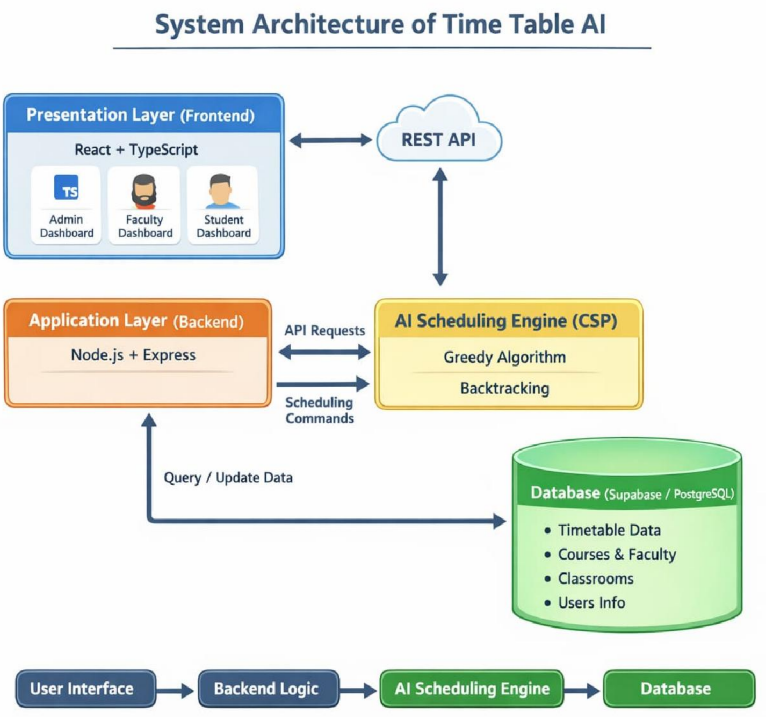
Early computational solutions used techniques like Integer Linear Programming (ILP) and Graph Coloring. These methods modeled timetabling as an optimization problem. They aimed to minimize constraint violations

DOI: 10.48175/IJARSCT-33413

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 10, April 2026
Impact Factor: 8.2

mathematically. While effective for small datasets, they lacked scalability. As problem size increased, computation became expensive and slow.

To overcome these limitations, researchers introduced meta-heuristic algorithms. These include Genetic Algorithms, Simulated Annealing, and Tabu Search. Genetic Algorithms explore large solution spaces efficiently. Simulated Annealing helps escape local optima. However, these methods require tuning and may not always guarantee feasible solutions.



A more recent approach models timetabling as a Constraint Satisfaction Problem (CSP). CSP focuses onsatisfying constraints rather than optimizing a cost function. It distinguishes between hard andsoft constraints. Backtracking is a key technique used in CSP. It systematically searches for valid solutions. Hybrid approaches combine backtracking with greedy heuristics. This improves efficiency and performance.

Modern research also emphasizes system design and usability. Web-based systems allow real-time access and updates. They support multiple users like administrators, faculty, and students. Features like lab batch rotation are also considered. These advancements make systems more practical and user-friendly.

In conclusion, timetabling research has evolved significantly over time. It has progressed from manual methods to advanced intelligent systems. CSP-based hybrid approaches offer reliable and scalable solutions. Time Table AI builds on these concepts using modern technologies. It provides an efficient and practical solution for academic scheduling.

III. PROBLEM STATEMENT

To address the combinatorial complexity and scalability challenges involved in managing numerous courses, faculty members, classrooms, and student batches, and to eliminate scheduling conflicts such as double-booking and overlapping sessions. To improve resource utilization by optimizing both hard and soft constraints, including room allocation and balanced faculty workload. To simplify and automate practical and laboratory scheduling with efficient batch rotation management. To reduce the time-consuming and labor-intensive nature of manual timetable creation

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 10, April 2026
Impact Factor: 8.2

while enabling flexibility for real-time updates and changes. To enhance accessibility and communication by providing a transparent system that allows students and faculty to easily access schedules anytime.

IV. PROPOSED SOLUTION

The increasing complexity of academic scheduling highlights the need for an intelligent, automated, and scalable solution. Traditional timetable generation methods fail to efficiently handle large datasets, dynamic constraints, and real-time changes. To overcome these limitations, Time Table AI is proposed as a smart scheduling system that integrates artificial intelligence, constraint satisfaction techniques, and modern web technologies. The system is designed to generate optimized, conflict-free timetables while ensuring efficient resource utilization and user accessibility.

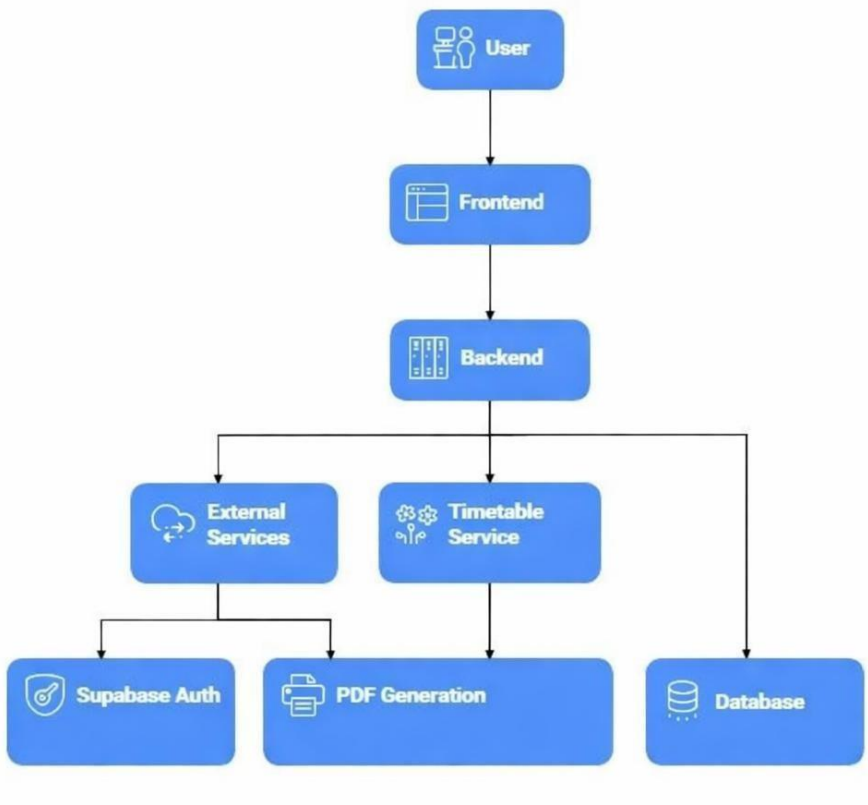


Figure 4.1: AI-Based Time Table Generation System

V. METHODOLOGY

The development of the Time Table AI system followed a structured and systematic methodology to ensure scalability and efficiency. The software stack was carefully selected to balance performance and developer productivity. The frontend was built using React with TypeScript to create a dynamic and type-safe user interface. React's component-based architecture enabled reusable UI elements, reducing code redundancy. TypeScript improved code reliability by preventing runtime errors.

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 10, April 2026 Impact Factor: 8.2

Vite was used as the build tool due to its fast development capabilities and Hot Module Replacement. TailwindCSS was adopted for consistent and modern UI design. Additional UI libraries like shadcn/ui and Radix UI enhanced accessibility and development speed.

The backend was developed using Node.js and Express.js to handle concurrent requests efficiently. Express.js provided a structured approach for API development and middleware management. Supabase with PostgreSQL was used as the database to manage relational data effectively. It also offered built-in authentication and security features.

The scheduling logic was implemented using custom algorithms for better control and transparency. Recharts was used for data visualization in analytics dashboards. Tools like html2canvas and jsPDF enabled PDF export functionality. Overall, the chosen technologies ensured a robust, scalable, and user-friendly system.

5.1 DATASET

The data management methodology in Time Table AI focuses on creating a dynamic and comprehensive dataset that reflects real-world academic operations. The system includes user and role-based data categorized into Admin, Faculty, and Student roles, each with specific attributes to support Role-Based Access Control (RBAC). Faculty data includes department and contact details, while student data includes semester and batch information.

The dataset also contains academic and curriculum data through a detailed course catalog, including course name, code, type, assigned faculty, department, semester, and weekly contact hours. This enables the system to distinguish between lectures, labs, and seminars

effectively. .

5.2 DESIGN

The design phase of Time Table AI focuses on building a scalable architecture and a well- structured database to support automated scheduling. The system follows a three-tier client-server architecture, consisting of presentation, application, and data layers. The frontend, built with React, handles user interaction and displays dashboards and timetables. The backend, developed using Node.js and Express, manages business logic and executes the scheduling algorithm. The data layer uses Supabase PostgreSQL for reliable and scalable storage. This separation ensures flexibility, maintainability, and independent scalability of each layer.

The database schema is designed using normalization principles to reduce redundancy and maintain data integrity. Key tables include users, courses, rooms, time_slots, and timetable, where the timetable table acts as a central link between all entities. Constraints are applied to prevent conflicts such as double booking, and indexing improves query performance for faster data retrieval.

The backend follows REST API principles, with endpoints structured around resources like courses and timetables. Middleware is implemented for authentication using JWT tokens and for role-based authorization, ensuring that only authorized users can perform sensitive operations.

The frontend adopts a component-based design, promoting modularity and reusability.

Components such as layouts and timetable views are reused across different user roles with different datasets.

The algorithmic flow is designed to be modular, with a clear separation between the scheduler and constraint checker. A recursive backtracking approach is used to ensure conflict-free scheduling. This design allows easy modification of rules and ensures the system remains adaptable to future institutional requirements.

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 10, April 2026
Impact Factor: 8.2

VI. RESULT

1. Login Page

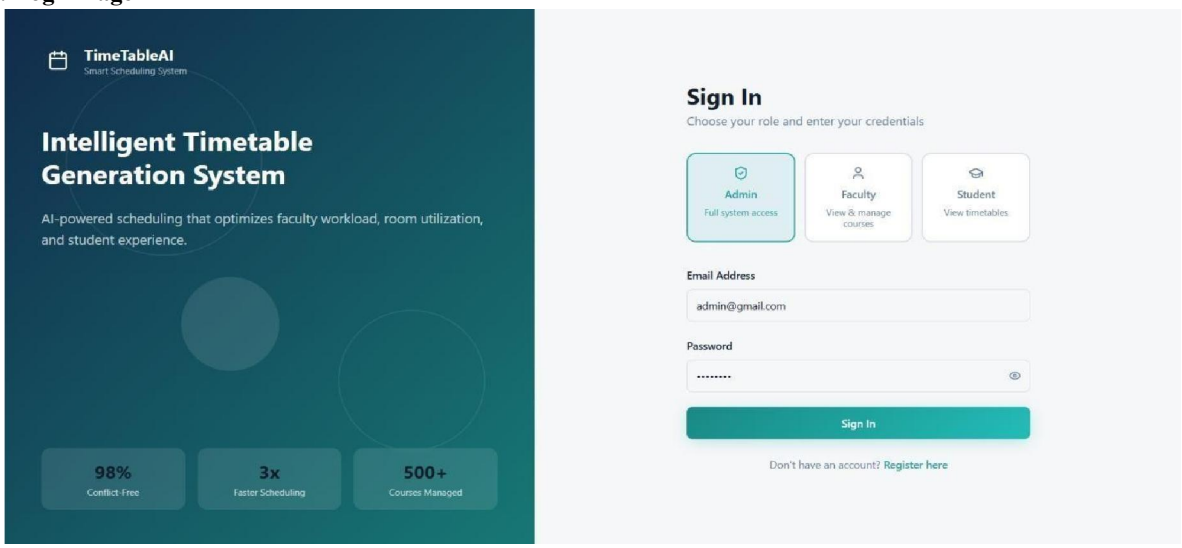


Figure 6.1: Login Page

Figure 6.1 shows the login page is the entry point to the Time Table AI system. It features a two-panel layout a branded left describing the system's purpose and key capabilities, and a sign- in form on the right. A role selection mechanism allows the user to identify as Admin, Faculty or Student before submitting credentials, ensuring they are directed to the appropriate dashboard upon login. The form includes email and password fields with registration link for new users.

2. Admin Dashboard

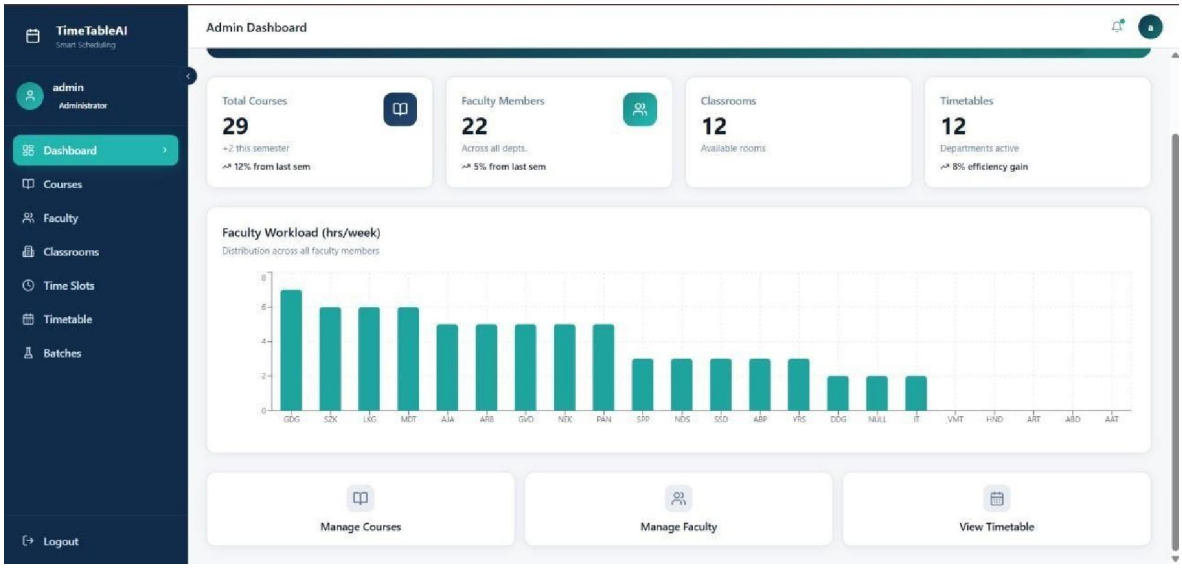


Figure 6.2: Admin Dashboard

Figure 6.2 shows the Admin Dashboard provides a high-level overview of the entire system. Summary stat cards at the top display live counts of key entities such as courses, faculty, classrooms and active timetables. A bar chart visualizes

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 10, April 2026
Impact Factor: 8.2

faculty workload distribution across all members, helping the admin identify, imbalances at a glance. Quick-action shortcut cards at the bottom fact navigation allow fast navigation to the most frequently used management sections.

3. Course Management Page

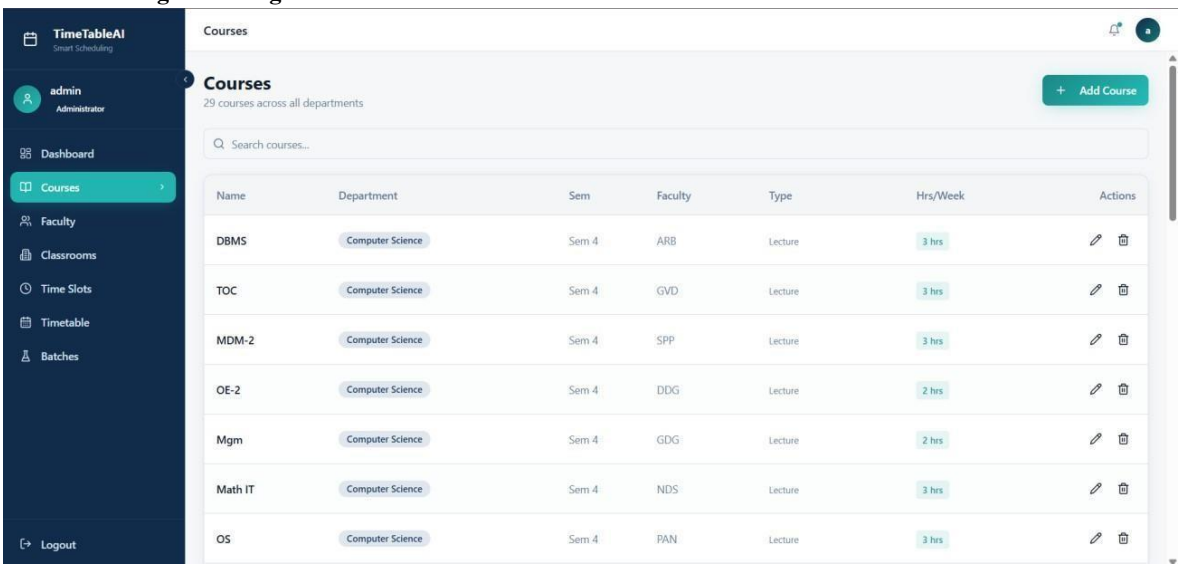


Figure 6.3: Course Management Page

Figure 6.3 shows the Courses page enables the administrator to manage the complete course catalog. Courses are listed in a tabular format showing relevant details such as name, department, semester assigned faculty, course type, and weekly hours. A search bar allows filtering by course name. Each entry includes edit and delete actions for inline management, and a button at the top allows adding new courses to the system.

4. Faculty Management Page

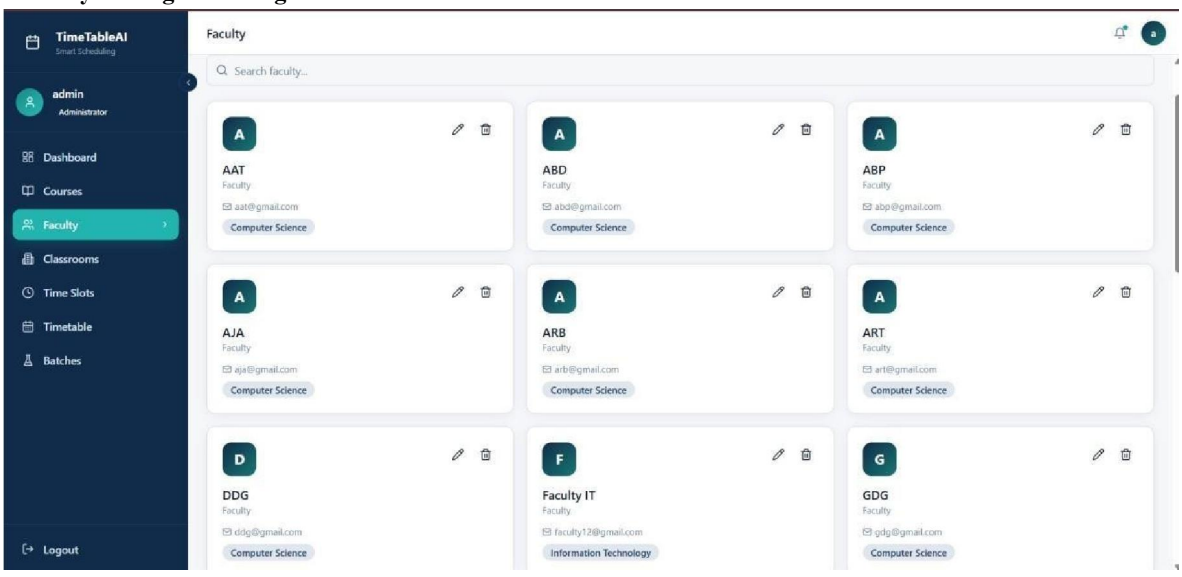


Figure 6.4: Faculty management page

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 10, April 2026
Impact Factor: 8.2

Figure 6.4 shows he Faculty Management page which displays all registered faculty members in a card-based grid layout. Each card represents the faculty member's basic information including name, role, email and department, A search bar enables filtering by name. Each card provides edit and delete actions, allowing the admin to manage faculty records directly from the view.

5. Classrooms Management Page

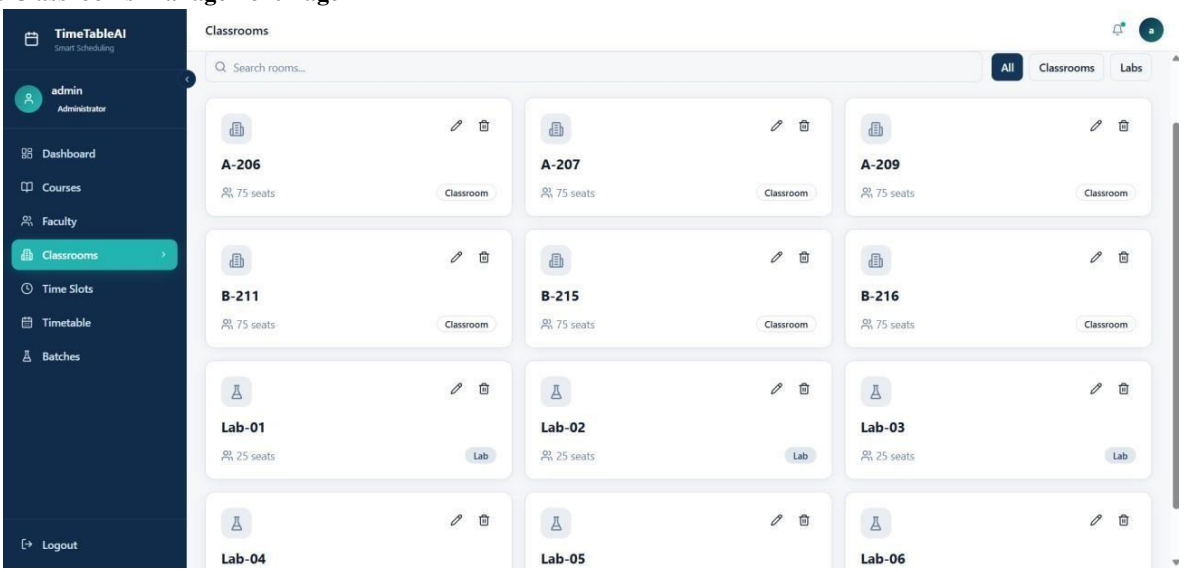


Figure 6.5: Classroom management page

Figure 6.5 shows the Classrooms page where it lists all available rooms in a card grid layout. Filter tabs at the top all the top allow admin to view all rooms or filter by type standard classrooms or laboratories. Each card displays the room name, seating capacity and room type badge. The visual distinction between classroom and lab types support the scheduling algorithm, which assign courses to rooms based on their typre requirements.

6. Timetable View (Admin)

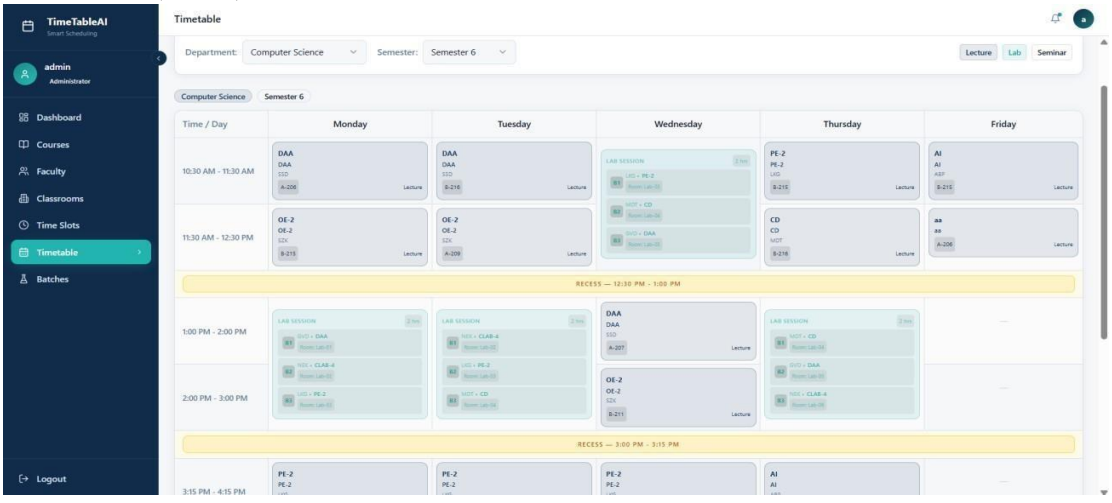


Figure 6.6: Table view (Admin)

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 10, April 2026
Impact Factor: 8.2

Figure 6.6 shows the Timetable page which displays the AI-generated weekly schedule for a selected department and semester. The schedule is rendered as a grid with time slots on the vertical axis and days of the week on the horizontal axis. Filter buttons allow toggling the visibility of lecture, lab and seminar slots. Lecture slots display course, faculty and room details, while lab sessions are visually highlighted and show batch-wise room assignment. Recess periods are clearly marked across the grid. The layout confirms that the algorithm has produced a conflict free, well distributed schedule.

7. Student Dashboard

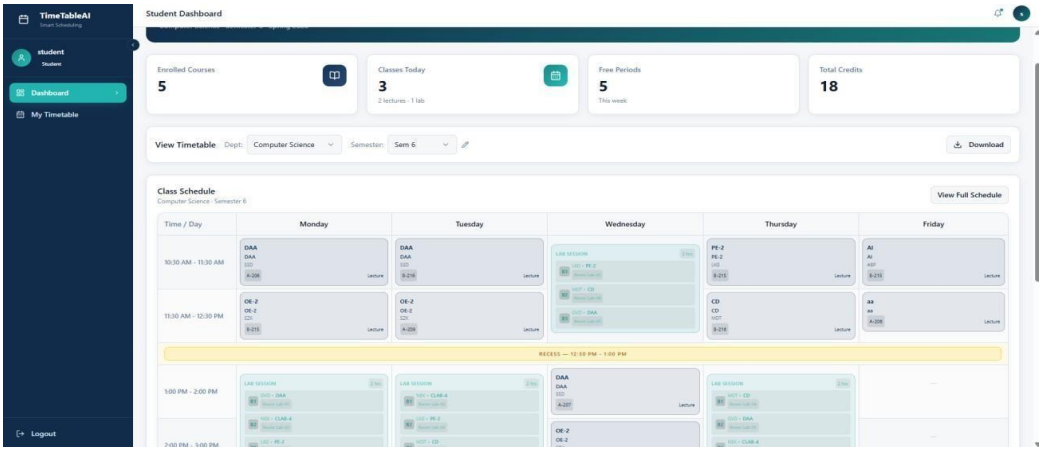


Figure 6.7: Student dashboard

Figure 6.7 shows the Student Dashboard which provides a personalized academic view for the logged-in student. Summary stat cards at the top show the student's enrolled courses, classes scheduled for the day, free periods, and total credits. A timetable viewer is embedded directly on the dashboard, allowing the student to view their department and semester schedule without navigating away. Lab sessions display the student's specific batch assignment, showing which lab course their batch attends and in which room. A download option allows exporting the schedule as a PDF.

8. Supabase Database Table Editor

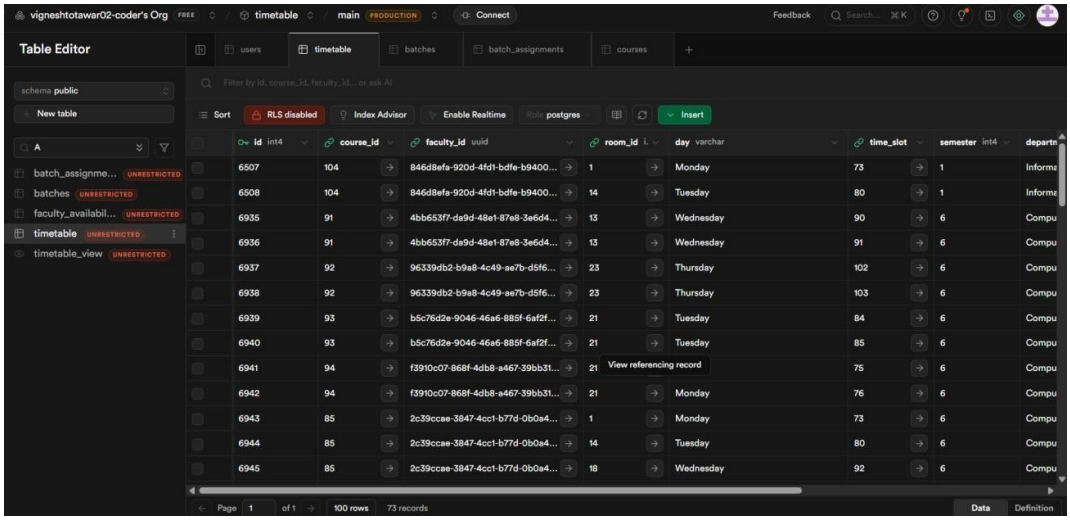


Figure 6.8: Supabase database table editor

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 10, April 2026 Impact Factor: 8.2

Figure 6.8 shows the Supabase Table Editor which provides a visual interface for managing the project's PostgreSQL database. The left sidebar lists all tables in the public schema, representing the core data entities of the system. The main panel displays table records in a spreadsheet-style grid, showing the relational fields that link courses, faculty, rooms, time slots, and scheduling data together. The interface confirms that the database is actively populated with records generated by the scheduling algorithm, and provides tools for sorting, filtering, and inserting data directly.

VII. CONCLUSION

The Time Table AI project successfully addresses the critical challenges of academic timetabling by replacing manual, error-prone processes with a robust, AI-driven solution. By leveraging a Constraint Satisfaction Problem (CSP) algorithm and modern web technologies, the system automates the generation of conflict-free schedules while optimizing resource utilization and workload distribution. The application effectively meets the diverse needs of administrators, faculty, and students through its role-based architecture and intuitive interface. Ultimately, Time Table AI serves as a comprehensive tool that eliminates scheduling bottlenecks, ensures operational transparency, and significantly enhances the administrative efficiency of educational institutions.

VIII. FUTURE SCOPE

The future scope of Time Table AI includes integrating advanced AI and machine learning models such as Genetic Algorithms and Simulated Annealing to improve optimization for large- scale datasets. The system can be enhanced with native mobile applications for Android and iOS to provide better accessibility, offline access, and push notifications. A real-time notification system can be implemented to send instant alerts via email, SMS, or messaging platforms for schedule changes.

Integration with institutional ERP and Student Information Systems can automate data synchronization and reduce manual data entry. The system can be expanded to support examination timetabling and event management, making it a complete resource management platform. Future improvements can also include enhanced soft constraint handling through preference-based scheduling for faculty and students.

Predictive analytics can be incorporated to forecast resource requirements and support strategic planning. Finally, the system can evolve into a cloud-based SaaS platform, allowing multiple institutions to use it efficiently with scalable and secure infrastructure.

REFERENCES

[1] E. K. Burke, S. Petrovic, and R. Qu, “Meta-heuristics for university timetabling problems,” Journal of Heuristics, vol. 12, no. 4–5, pp. 347–368, 2006.

[2] R. Lewis, “A survey of metaheuristic-based techniques for university timetabling problems,” Omega, vol. 36, no. 1, pp. 39–53, 2008.

[3] A. Schaerf, “A survey of automated timetabling,” Artificial Intelligence Review, vol. 13, no. 2, pp. 87– 127, 1999.

[4] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Pearson Education, 2020.

[5] R. Dechter, Constraint Processing. Morgan Kaufmann, 2003.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, 3rd ed. MIT Press, 2009.

[7] H. Babaei, J. Karimpour, and A. Hadidi, “A survey of approaches for university course timetabling problem,” Computers & Industrial Engineering, vol. 86, pp. 43–59, 2015.

[8] B. McCollum, P. McMullan, A. J. Parkes, E. K. Burke, and R. Qu, “A new model for timetabling problems,” in Practice and Theory of Automated Timetabling (PATAT), pp. 491–493, 2010.

[9] N. Pillay and W. Banzhaf, “An informed genetic algorithm for the examination timetabling problem,” Applied Soft Computing, vol. 10, no. 2, pp. 457–467, 2010.

[10] S. B. Deris, S. Omatu, H. Ohta, and L. C. Shaharudin, “University timetabling using constraint-based reasoning: A case study,” Knowledge-Based Systems, vol. 12, no. 5–6, pp. 319–327, 1999.

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 10, April 2026
Impact Factor: 8.2

[11] S. Even, A. Itai, and A. Shamir, "On the complexity of timetable and multicommodity flow problems," SIAM Journal on Computing, vol. 5, no. 4, pp. 691–703, 1976.
[12] Meta Platforms, Inc., "React Documentation: A JavaScript library for building user interfaces," 2024. [Online]. Available: https://react.dev
[13] OpenJS Foundation, "Node.js Documentation," 2024. [Online]. Available: https://nodejs.org/docs
[14] Express.js Foundation, "Express.js Documentation," 2024. [Online]. Available: https://expressjs.com
[15] Supabase Inc., "Supabase Documentation: The open-source Firebase alternative," 2024. [Online]. Available: https://supabase.com/docs
[16] PostgreSQL Global Development Group, "PostgreSQL 16 Documentation," 2024. [Online]. Available: https://www.postgresql.org/docs