Smart Classroom and Timetable Scheduler

Kundan Kumar giri[1], Saurabh Lode[2], Jayshree Bagadiya[3]

ABSTRACT:

Keywords: Smart Classroom, Automated Timetable Scheduling, Timetable Generator, Classroom Management System, Scheduling Algorithm, Conflict-Free Scheduling, Educational Technology, Resource Optimization, Academic Management System, CSV Data Processing

Creating a timetable is one of the most important and challenging administrative tasks in educational institutions. It involves coordinating various factors such as teachers, subjects, class timings, and institutional requirements. Preparing timetables manually takes a lot of time and often leads to errors, including overlapping classes, overloading teachers, and poor distribution of subjects. As the number of students and subjects continues to grow, the traditional manual method is becoming increasingly difficult to manage and less practical.

This project focuses on designing and implementing an Automatic Timetable Generator that creates a valid and conflict-free timetable using structured data from *teachers.csv* and *subjects.csv* files. The system reads the relationships between teachers and subjects, considers scheduling constraints, and assigns subjects to appropriate time slots. While doing so, it ensures that no teacher is scheduled for more than one class at the same time. The final timetable is presented in a clear and well-organized tabular format, making it easy to understand and maintain for future reference.

The proposed system helps reduce manual effort, removes scheduling conflicts, and improves overall efficiency in timetable creation. It is cost-effective, user-friendly, and can be easily adopted by schools, colleges, and training institutes. In addition, this project provides a strong base for future improvements, such as developing a web-based version, adding room allocation features, and applying advanced optimization techniques.

I. INTRODUCTION

In every educational institution, the timetable plays a central role in managing academic activities. It determines how subjects, teachers, and students are arranged across different days and time periods. A well-planned timetable helps in balancing workloads, making efficient use of resources, and ensuring smooth academic operations. However, creating such a timetable manually is both time-consuming and complex due to the many constraints involved.

Traditionally, academic coordinators or administrators prepare timetables manually, which often requires repeated checking, adjustments, and modifications to avoid conflicts. Even minor errors, such as assigning a teacher to two classes at the same time, can cause significant disruptions. As the number of departments, teachers, and subjects increases, the process becomes even more complicated and difficult to manage.

This project aims to develop an Automatic Timetable Generator that uses input data from CSV (Comma-Separated Values) files, specifically *teachers.csv* and *subjects.csv*. CSV files are commonly used because they are simple, portable, and compatible with a wide range of software platforms. By using CSV files as input, the system makes it easy to update data without needing to change the core application.

The main objective of this project is to simplify the timetable creation process, reduce the workload on administrators, and generate a conflict-free schedule efficiently. The system is designed to be simple, scalable, and suitable for practical use in real-world academic environments.

II. PROBLEM STATEMENT

In many educational institutions, classroom management and timetable scheduling are still performed manually or with basic tools such as spreadsheets, which often results in inefficiencies and errors. This traditional approach frequently leads to scheduling conflicts where teachers are assigned to multiple classes at the same time, classrooms are double-booked, and subject distribution becomes unbalanced across the week. Additionally, manual timetable preparation increases the administrative workload and makes it difficult to implement quick changes when unexpected situations arise. Existing digital platforms such as Moodle and Google Classroom mainly focus on content sharing and communication, but they do not provide integrated solutions for automated timetable scheduling and classroom resource management. Furthermore, there is a lack of synchronization between various academic systems like attendance tracking, assignment management, and scheduling, leading to data duplication and reduced efficiency. As institutions grow larger, managing multiple classrooms, subjects, and teachers becomes increasingly complex, making it difficult to ensure that all constraints are satisfied, such as avoiding teacher clashes, proper classroom allocation, and fulfilling required teaching hours. Therefore, there is a strong need for an integrated Smart Classroom Management and Automated Timetable Scheduling System that can reduce conflicts, optimize resource utilization, minimize manual effort, and improve the overall efficiency of academic operations.

Manual timetable preparation often leads to problems such as:

Teacher time clashes

Classroom allocation conflicts

Unequal subject distribution

Improper workload balancing

Time-consuming modifications during sudden changes

Additionally, existing digital platforms such as Moodle and Google Classroom mainly focus on content delivery and communication, but they do not offer integrated features for automated timetable scheduling along with classroom resource management.

Another key issue is the lack of integration between classroom management systems and scheduling systems. In most institutions, separate tools are used for tasks like attendance tracking, assignment management, and timetable preparation. This separation often results in data duplication and reduces overall efficiency.

Furthermore, as institutions grow in size, managing multiple classrooms, teachers, subjects, and laboratory sessions becomes increasingly complex. Without an optimized scheduling mechanism, it becomes difficult to satisfy all constraints such as:

No teacher should be assigned to two classes at the same time.

A classroom should not be allocated to multiple classes simultaneously.

Required subject hours per week must be fulfilled.

Laboratory sessions require special room allocation.

Therefore, there is a need for an integrated Smart Classroom Management and Automated Timetable Scheduling System that can:

Reduce scheduling conflicts

Optimize resource allocation

Minimize manual effort

Improve transparency and efficiency

Provide centralized management for administrators, teachers, and students

This project aims to design and implement a system that combines smart classroom features with automated timetable generation to enhance the overall academic management process.

III. LITERATURE REVIEW

The rapid growth of digital technology has brought significant changes to the education system. Traditional methods of classroom management and manual timetable scheduling come with several limitations, such as time conflicts, inefficient use of resources, and increased administrative workload. To address these challenges, many researchers have explored smart classroom systems and automated scheduling techniques as effective solutions.

Smart classroom systems make use of digital tools such as interactive boards, projectors, online attendance systems, and learning management platforms. Platforms like Moodle and Google Classroom allow teachers and students to share content, submit assignments, and communicate easily. While these systems enhance accessibility and improve the overall learning experience, they do not completely address issues related to timetable optimization and effective classroom resource management.

Timetable scheduling has been widely studied as a complex optimization problem. Manual methods of scheduling often result in issues such as teacher clashes, classroom conflicts, and uneven distribution of subjects. Researchers have classified timetable generation as an NP-hard problem, which means it requires careful handling of multiple constraints to produce efficient and practical solutions.

IV. METHODOLOGY

4.1. Research Approach

The development of the Smart Classroom Management and Automated Timetable Scheduling System follows a structured software development approach. The methodology includes requirement analysis, system design, implementation, testing, and evaluation.

A modular development strategy is adopted to ensure that each component such as user management, timetable generation, and classroom management functions independently and efficiently.

4.2. System Architecture

The Automatic Timetable Generator follows a modular architecture consisting of:

Input Module: The input module reads data from teachers.csv and subjects.csv files.

Processing Module: The processing module cleans and structures the data.

Scheduling Module: The scheduling module applies allocation logic and conflict-checking rules.

Output Module: The output module generates and displays the final timetable.

4.3. Requirement Analysis

In this phase, the functional and non-functional requirements of the system were identified.

Functional Requirements:

Admin should be able to add teachers, students, subjects, and classrooms.

System should generate timetable automatically.

Teachers should view timetable and mark attendance.

Students should view timetable and submit assignments.

System should prevent time and classroom clashes.

Non-Functional Requirements:

System should be user-friendly.

Data should be stored securely.

System should respond quickly.

It should support multiple users simultaneously.

4.3. Tools and technologies

The development of the Smart Classroom Management and Automated Timetable Scheduling System incorporates a combination of frontend, backend, database, authentication mechanisms, and external service integrations. The details are as follows:

Frontend Technologies

HTML (Hyper Text Markup Language): Used for structuring web pages, with *index.html* serving as the main entry point of the application.

CSS (Cascading Style Sheets): Employed for styling and layout design through files such as *styles.css* and *module.css*.

Vanilla JavaScript: Implements client-side logic and dynamic behavior using scripts like *app.js* and *module.js*.

Chart.js: Integrated to visualize data through interactive charts and dashboards.

Google Fonts: The *Outfit* font is used to enhance the visual appearance and readability of the user interface.

Backend Technologies

The backend is responsible for handling business logic, API communication, and server-side processing:

Node.js: Serves as the runtime environment for executing server-side JavaScript code.

Core HTTP Module: The backend is developed using Node.js's native HTTP module instead of frameworks like Express, ensuring lightweight and customizable server implementation.

Common JS (require): Module system used for structuring backend code.

Doten v: Facilitates secure configuration management by loading environment variables from a *.env* file.

crypto Module: Used for implementing secure authentication mechanisms, including token generation and signature validation.

https Module: Enables communication with external APIs over secure protocols.

Database and Storage

The system supports multiple database options to ensure flexibility and scalability:

JSON File Database: A lightweight default storage mechanism using for quick setup and testing.

MongoDB: A NoSQL database option for scalable and flexible data storage.

MySQL: A relational database option for structured data management.

The database provider is configurable through environment variables (*DB_PROVIDER*), allowing seamless switching between JSON, MongoDB, and MySQL.

Authentication and Authorization

The system implements secure and role-based access control:

Role-Based Authentication: Supports multiple user roles such as Admin, Teacher, and Student.

Custom Token-Based Authentication: Utilizes cryptographic techniques (HMAC and tokens) for secure session management.

Google OAuth Integration: Optional login functionality using *google-auth-library* for authentication via Google accounts.

During the design of the Automatic Timetable Generator, special attention was given to reducing both computational and manual effort while improving scheduling efficiency. The development process included proper input validation, efficient data handling using CSV files, and the use of a lightweight scheduling algorithm to ensure fast execution.

To assess the effectiveness of the proposed approach, an analysis of Estimated Effort versus Efficiency Achieved was carried out. In this context, estimated effort refers to the amount of computational processing and manual setup required, while efficiency is evaluated based on factors such as time saved, reduction in scheduling conflicts, and accurate allocation of subjects.

As illustrated in Fig.1, the proposed system demonstrates a significant improvement in efficiency compared to traditional manual scheduling even with a relatively low computational effort. This analysis validates the practicality and scalability of the adopted methodology for real-world academic environments.

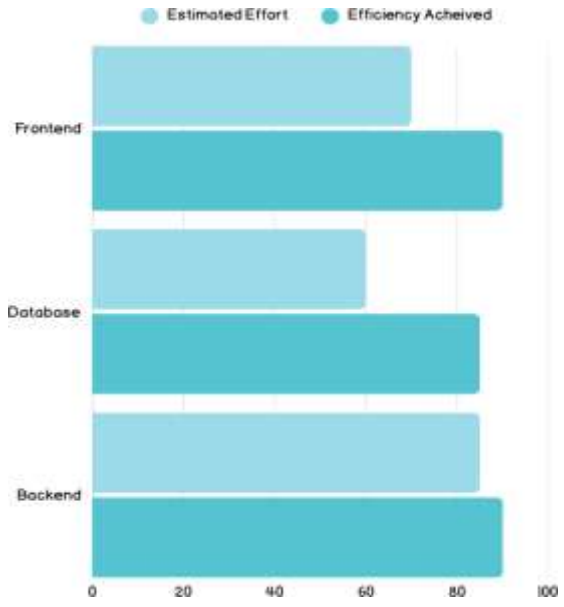

Figure1: Estimated effort vs Efficiency Achieved

V. FUTURE SCOPE

The proposed Smart Classroom Management and Automated Timetable Scheduling System provides an efficient solution for managing classroom activities and timetable generation. However, the system can be further enhanced with advanced technologies and additional features in the future.

5.1. Integrating with Artificial Intelligence

In the future, Artificial Intelligence (AI) can be integrated to improve timetable optimization. AI-based algorithms such as machine learning models can analyze historical scheduling data and automatically generate more efficient and conflict-free timetables. This would further reduce manual intervention and improve workload distribution among teachers.

5.2. Mobile Application Development

The system can be extended into a mobile application for Android and iOS platforms. A dedicated mobile app would allow students and teachers to:

Receive real-time notifications

View updated timetables instantly

Mark attendance through mobile devices

Submit assignments easily

This would improve accessibility and user engagement.

VI. RESULT AND OUTPUT

The Automatic Timetable Generator was tested using multiple sample datasets containing different combinations of teachers and subjects as shown in Fig.2 below.

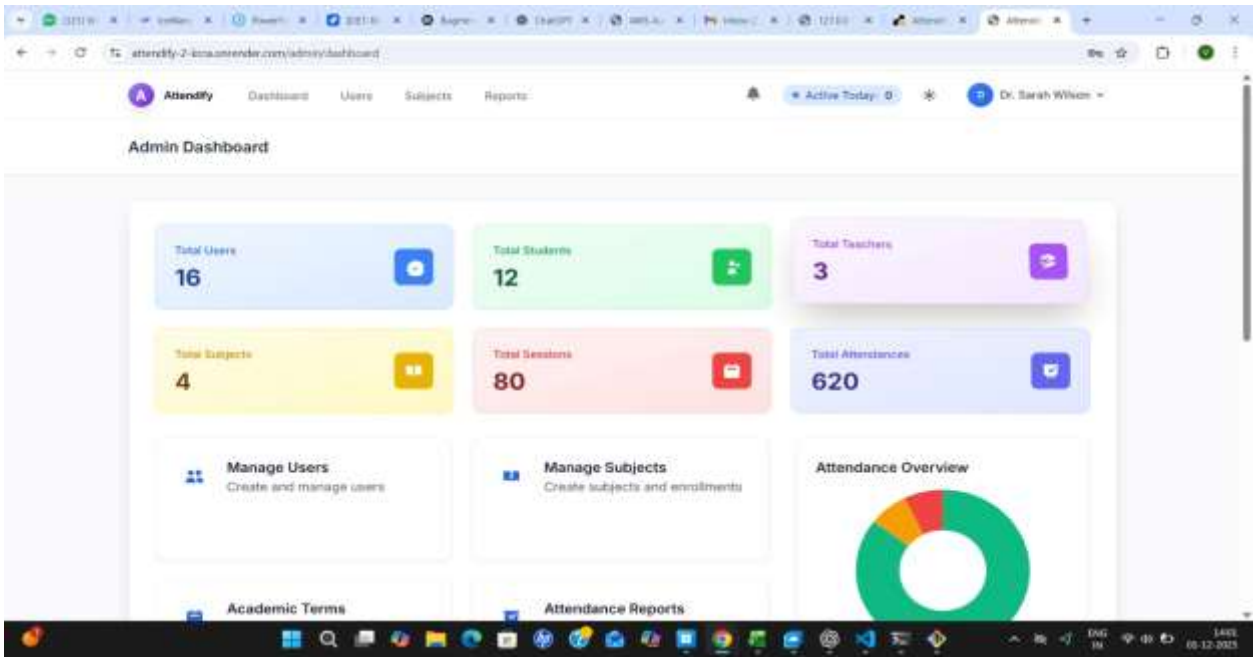

Figure 2: output of the homepage

The system successfully generated weekly timetables without any teacher conflicts. The output timetable as shown in Fig.3 was displayed in a structured tabular format showing days as rows and periods as columns.

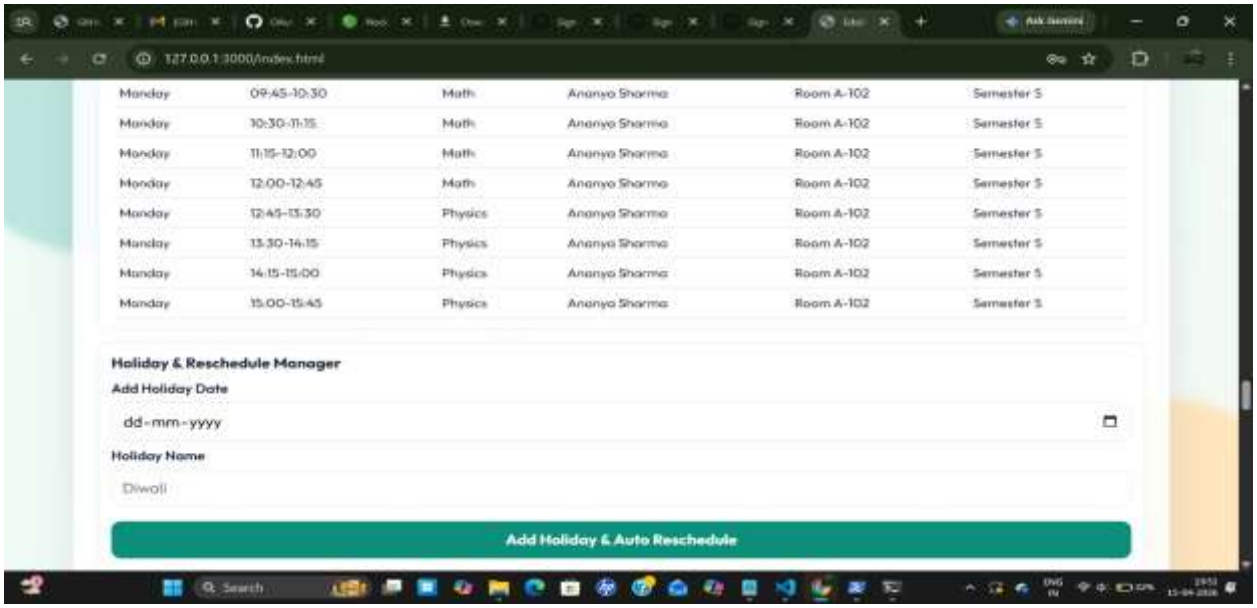

Figure 3: output of the generated timetable

VII. DISCUSSION

Each subject was assigned based on the expertise of the respective teacher. If a teacher was unavailable during a specific period, the system marked that slot as free, ensuring clarity and transparency in the schedule. It also ensured that no teacher was allocated more than one subject at the same time.

Performance testing indicated that a complete weekly timetable could be generated within a few seconds, even when handling multiple teachers and subjects. The system remained stable and consistently produced accurate results for all valid input files. Additionally, the generated timetable can be easily exported, printed, or stored digitally for future use. The system is also flexible, allowing updates to teacher or subject information simply by modifying the CSV files.

The development of the Smart Classroom Management and Automated Timetable Scheduling System highlights how digital transformation can enhance academic administration and improve classroom efficiency. The results of the implemented system show clear improvements over traditional manual scheduling methods.

One of the major challenges faced by educational institutions is managing timetable conflicts. Manual scheduling often leads to issues such as teacher clashes, classroom overlaps, and improper distribution of subject hours. The proposed system effectively addresses these problems by using constraint-based scheduling techniques, ensuring that all predefined conditions are met before generating the final timetable.

In addition, the system improves resource utilization. By automatically monitoring teacher availability and classroom allocation, it minimizes idle time and ensures a balanced distribution of workload. This leads to better time management and increased institutional productivity.

Overall, the system demonstrates that integrating automated scheduling with smart classroom management can greatly enhance efficiency, reduce human errors, and improve transparency in academic operations. The discussion also suggests that future enhancements, such as incorporating artificial intelligence, cloud-based deployment, and data analytics, can further improve the system's scalability and intelligence.

VIII. CONCLUSION

The Smart Classroom Management and Automated Timetable Scheduling System was developed to overcome the limitations of traditional classroom administration and manual timetable preparation. Educational institutions often face challenges such as scheduling conflicts, inefficient use of resources, increased administrative workload, and the absence of a centralized management system. This project presents a solution that combines smart classroom features with automated timetable generation.

The system uses constraint-based scheduling techniques to avoid issues like teacher time clashes, classroom conflicts, and uneven distribution of subject hours. By automating the timetable creation process, it reduces manual effort and minimizes the chances of human error. In addition, features such as attendance management, assignment handling, and centralized notifications help improve communication and ensure better transparency among administrators, teachers, and students.

Integration of a web-based interface

Student batch and classroom allocation

Laboratory and practical session scheduling

Teacher absence and availability management

AI-based optimization for large-scale institutions

Real-time timetable updates

IX. REFERENCE

- Bagul, M. R., Chaudhari, S. C., S. N., Patil, P. R., K. S. (2015). *A Novel Approach for Automatic Timetable Generation*. International Journal of Computer Applications, 127(10), 26–30.
- Tank Sali, P., Dhond, I., Pednekar, S, V., & Sivaraman, S. (2020). *Automated Timetable Generation*. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 6(3).
- Prasad Reddy, K. V Krishna, B., Nithin Sai, T., Surekha, Y. T., Kowsalya, G., & Reddy, K. (2021). *An Automatic Time Table Generation*. International Journal of Scientific Research in Science and Technology (IJSRST), 8(3).
- Thakur, R. K., Agrawal, N. K., & Kumar, P. (2024). *A Practical Approach to College Timetable Scheduling*. Mathematical Modeling and Computing, 11(3), 710–719.

- Rahman, M. M., Noor, S. B., & Siddiqui, F. H. (2020). *Automated Large-scale Class Scheduling in*.
- Hooshmand, S., h, M., & Hamidi, O. (2013). *A Tabu Search Algorithm with Efficient Diversification Strategy for High School Timetabling Problem* Preprint.
- *From Integer Programming to Machine Learning: A Technical Review on Solving University Timetabling Problems*. (2023). *Computers*, MDPI.
- *Student Timetabling Genetic Algorithm Accounting for Student Preferences*