International Journal of
Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 4, 2025

Smart Timetable Suggestion App

[1]Akula Lakshmi Priyanka, [2]Chelli Asha Jyothi, [3]Ambati Pavan Tanwesh, [4]Mrs. V L N. Swapnika

[1,2,3]U. G Student, Dept COMPUTER SCIENCE AND ENGINEERING, St.Ann's College Of Engineering and Technology, Nayunipalli (V), Vetapalem (M), Chirala, Bapatla Dist, Andhra Pradesh – 523187, India

[4]Assistant professor, COMPUTER SCIENCE AND ENGINEERING, St.Ann's College Of Engineering and Technology, Nayunipalli (V), Vetapalem (M), Chirala, Bapatla Dist, Andhra Pradesh – 523187, India

ABSTRACT

Creating academic schedules manually in educational institutions often results in class conflicts, errors, and inefficient use of faculty and classroom resources. To address these issues, the Smart Timetable Suggestion App is developed as a web- based solution capable of automatically generating accurate, conflict-free, and well-organized schedules. The system utilizes the MERN stack (MongoDB, Express.js, React.js, Node.js) to manage data, backend logic, and UI rendering. The application considers teacher availability, subject load, and classroom constraints to suggest optimized timetables. By shifting from conventional manual scheduling to an automated approach, the system saves time, reduces workload, minimizes human error, and enhances scheduling efficiency for administrators, faculty, and students.

Keywords

Smart Timetable Generation, Automated Scheduling, MERN Stack, MongoDB, Express.js, React.js, Node.js, Web-Based Scheduling System, Conflict-Free Timetable, AI-Based Scheduling, Time Management Automation, Academic Scheduling, Optimization Techniques, User-Friendly Interface, Data-Driven Timetable, Educational Technology.

INTRODUCTION

Scheduling in academic institutions is a complex and time-consuming task, especially when conducted manually. Traditional timetable creation often leads to overlapping classes, mistakes, and suboptimal allocation of classrooms and faculty. As educational institutions continue to grow, the demand for automated and efficient scheduling systems has significantly increased. The Smart Timetable Suggestion App aims to overcome the limitations of manual scheduling by utilizing modern web technologies. Developed using the MERN stack, the system stores faculty and subject information in MongoDB, uses Node.js and Express.js for backend communication, and provides an interactive front-end using React.js. The application intelligently processes data on teacher availability, subject requirements, and classroom capacities to generate well-structured timetables that reduce conflicts and improve resource utilization. This transition to automated scheduling enhances accuracy, saves time, and offers a more reliable solution for institutions.

LITERATURE REVIEW

Several studies have focused on automating academic scheduling using various computational and AI-driven techniques. Sahith Siddharth Paramatmuni et al. (2024) proposed a genetic algorithm-based timetable generator that optimizes schedules by analyzing faculty availability and classroom limits. Sunday J. Agbolade et al. (2024) introduced a hybrid genetic- greedy algorithm, demonstrating improved efficiency in generating complex university exam timetables. A survey study titled "From Integer Programming to Machine Learning: A Technical Review on Solving University Timetabling Problems" (MDPI, 2023) discussed the increased adoption of AI, optimization algorithms, and hybrid approaches for scalable scheduling solutions. These studies highlight the growing need for smarter systems that utilize algorithms and modern technologies to automate scheduling tasks effectively, forming the foundation for the proposed Smart Timetable Suggestion App.

RELATED WORK

Many educational institutions still rely on manual timetable creation using paper or Excel sheets. Although some automated systems exist, they often struggle with updates, scalability, and handling complex constraints. Earlier systems lacked user- friendly interfaces and failed to adapt to frequent schedule changes. Compared to these traditional and outdated approaches, the Smart Timetable Suggestion App uses the MERN stack to build a flexible, scalable, and interactive scheduling system. MongoDB stores subject and teacher data, Node.js and Express.js handle backend processing, while React.js provides a dynamic user interface. This modern approach allows faster updates, improved accuracy, conflict-free scheduling, and better collaboration among administrators, teachers, and students.

EXISTING METHOD

Traditional timetable creation methods rely heavily on manual work using paper documents or spreadsheet tools such as Excel. These methods are often: Time- consuming Error-prone Difficult to update or modify Inefficient when handling frequent changes Challenging to optimize for resource allocation Some existing automated systems use genetic algorithms, but these require proper configuration and
can be slow in generating optimal solutions. Additionally, many current tools have confusing interfaces, making them less useful for teachers and students.

PROPOSED METHOD

The proposed Smart Timetable Suggestion App automates the entire scheduling process using modern web technologies. Built with the MERN stack, the system efficiently gathers and processes information such as teacher availability, subject details, and class hours to generate conflict-free timetables. Key features include: Automatic generation of optimized timetables User-friendly dashboard for administrators Database integration to securely store and retrieve data Responsive design to access schedules on any device Easy modification and updates This approach reduces workload, enhances accuracy, and saves time while providing a modern and scalable solution for educational institutions.

SYSTEM ARCHITECTURE

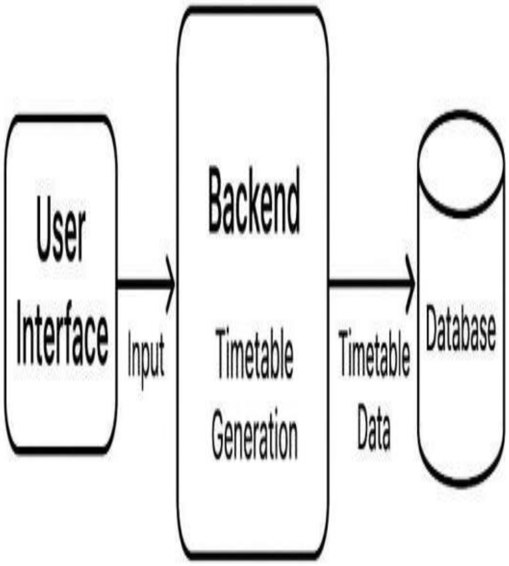


Fig1: Architecture of Smart Timetable Suggestion App

METHDOLOGYDESCRIPTION

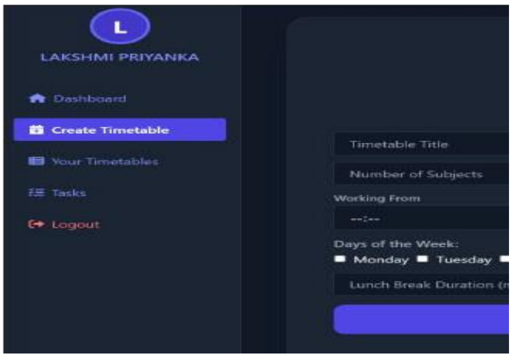
The methodology involves a sequential flow across three main components: User Interface, Backend, and Database.

1. User Interface (React.js):
   Users enter inputs including subjects, available time slots, workload, and teacher preferences using an intuitive web form.
2. Backend (Node.js / Express.js or Flask in referenced architecture):
   The backend processes the input data, checks for conflicts, and applies logic to generate an optimized timetable. It ensures compliance with user constraints and resource limitations.
3. Database (MongoDB / other storage): The generated timetable, user data, and system configurations are securely stored. Future timetable retrievals and modifications are performed through database interactions.
4. Output Generation:
   Users can view the generated timetable on the interface, download it as a PDF, and save it for future reference.
   This seamless integration ensures effective timetable creation with minimal manual intervention.

RESULTS AND DISCUSSION

Fig2: Home page for Smart Time Table Suggestion App

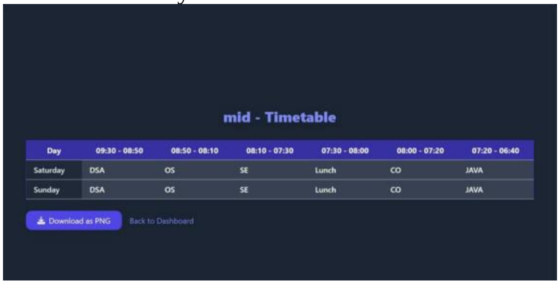
The Home Page of the Smart Timetable Suggestion App serves as the main entry point for users, providing an overview of the application's purpose and features. The timetable generation feature in the Smart Timetable Suggestion App automatically creates a personalized schedule based on the user's inputs, such as subjects, available time slots, and preferences. It utilizes AI logic (via the Cohere API) to suggest an optimized and balanced timetable that efficiently manages the user's workload. The generated timetable is neatly displayed on the results page and can also be downloaded as a PDF using ReportLab, ensuring easy access and printing for the user.



Timetable Title
Number of Subjects
Working From
--:--
Days of the Week:
■ Monday ■ Tuesday ■
Lunch Break Duration (r

Fig3: Time Table inputs

The observed outputs of the Smart Timetable Suggestion App include a well- structured and visually clear timetable generated based on the user's inputs. The application successfully displays the timetable on the results page, presenting subjects, timings, and days in an organized and easily readable layout. Additionally, the system offers useful features such as saving generated timetables, viewing previously created schedules, and downloading them in PDF format. Overall, the outputs demonstrate accurate scheduling, efficient functionality, and a user-friendly presentation of results.



mid - Timetable

Day	09:30 - 08:50	08:50 - 08:10	08:10 - 07:30	07:30 - 08:00	08:00 - 07:20	07:20 - 06:40
Saturday	DSA	OS	SE	Lunch	CO	JAVA
Sunday	DSA	OS	SE	Lunch	CO	JAVA

Fig4: Generated output

CONCLUSION

The Smart Timetable Suggestion App provides an efficient and intelligent solution for timetable creation. By automating the scheduling process using web technologies and AI logic, the app reduces manual effort, minimizes errors, and saves significant time. The user- friendly interface designed using HTML, CSS, JavaScript, and React.js ensures seamless user interactions. The backend, built with Flask or MERN tools, manages data processing and logic to produce optimized schedules. Overall, the system demonstrates how automation and AI can improve productivity and accuracy in everyday academic operations, making it useful for both students and faculty.

FUTURE SCOPE

Integration of advanced AI or machine learning models to predict optimal schedules. Addition of mobile applications for Android and iOS. Integration of automated notification systems for timetable changes. Support for multi- department scheduling and resource sharing. Cloud deployment for real-time multi-user access. Inclusion of analytics dashboards for monitoring faculty load and classroom utilization.

REFERENCES

[1] Ahmed, M., & Hassan, R. (2021). *Automated Timetable Generation Using Constraint Satisfaction Algorithms*. International Journal of Computer Applications, 183(45), 12–18.

[2] Kaur, S., & Sharma, A. (2020). *Smart Scheduling Systems Based on Machine Learning*. International Journal of Innovative Research in Computer and Communication Engineering, 8(6),
2235–2243.

[3] Patil, R., & Kulkarni, S. (2022). *AI- Based Timetable Management System Using Python and Flask*. Journal of Emerging Technologies and Innovative Research, 9(4), 121–128.

[4] Singh, R., & Gupta, M. (2021). *Optimization of College Timetables Using Genetic Algorithms*. Procedia Computer Science, 190, 256–262.

[5] Sultana, F., & Rahman, M. (2020). *Smart Scheduling and Task Management Using Web Technologies*. International Journal of Advanced Computer Science and Applications, 11(5), 77–83.

[6] Python Software Foundation. (2024). *Flask Documentation*. Retrieved from https://flask.palletsprojects.com

[7] Wirth, N. (2021). *Automated Timetable Generation with Python and SQLite*. IEEE Xplore Digital Library.

[8] Kumar, A., & Singh, P. (2019). *Intelligent Timetable Planner Using Heuristic Algorithms*. International Journal of Engineering Research and Technology, 8(9), 150–155.

[9] Thomas, D., & Joseph, S. (2021). *Smart Scheduling Using Web-based Frameworks*. International Journal of Web Applications, 13(2), 54–61.

[10] Rajesh, T., & Kumar, M. (2023). *Design and Implementation of a Flask- Based Timetable Automation System*. Journal of Software Engineering and Applications, 16(3), 75–84.

[11] Manna, D., & Ghosh, A. (2020). *Data-Driven Optimization in Scheduling Applications*. International Conference on Computational Intelligence, Springer, 201–209.

[12] Dash, P., & Mehta, S. (2021). *Automating Academic Scheduling Using AI Techniques*. Journal of Artificial Intelligence and Data Science, 2(4), 34–42.

[13] Das, R., & Nandi, P. (2022). *Smart Timetable and Task Allocation Using Flask and MySQL*. International Journal of Modern Engineering Research, 12(6), 93–99.

[14] Tiwari, K., & Verma, P. (2021). *Implementation of a Web-based Academic Management System*. International Journal of Computer Science and Mobile Computing, 10(7), 101–110.

[15] IBM Research. (2023). *Scheduling Optimization Techniques*. IBM Developer Documentation.

[16] Chandra, A., & Jain, V. (2020). *Hybrid Approach for Timetable Generation Using AI and Database Management*. Journal of Computer Applications, 184(12), 59–65.

[17] Stack Overflow Community. (2023). *Flask Application Structure for Data-Driven Web Apps*. Retrieved from https://stackoverflow.com

[18] Patel, J., & Shah, H. (2021). *College Timetable Scheduler Using Python and Web Technologies*. International Journal of Computer Trends and Technology, 69(8), 45–52.

[19] OpenAI. (2024). *Using AI to Automate Scheduling and Task Allocation*. Technical Whitepaper.

[20] World Academic Journal of Information Technology. (2022). *Design and Development of Smart Timetable Systems Using AI and Cloud Integration*.