AI-Based Timetable Generation System Aligned with NEP 2020

[1]Dr. D. Nagesh Babu,[2]Kumbha Akshaya,[3]Matcha Yaswanth Reddy,[4]Kahara Prince

[1]Associate Professor, Dept of Computer Science and Engineering, St. Ann's College of Engineering and Technology, Chirala-523187, India.

[2,3,4]U. G Student, Dept of Computer Science and Engineering, St. Ann's College of Engineering and Technology, Chirala-523187, India.

ABSTRACT- ***This project presents an AI-Based Timetable Generation System aligned with the National Education Policy (NEP) 2020. The system automates the process of generating academic timetables by considering faculty availability, classroom allocation, course requirements, student preferences, and institutional constraints. It improves efficiency, accuracy, and flexibility while supporting the multidisciplinary and student-centric approach promoted by NEP 2020. The system uses artificial intelligence and optimization techniques to generate conflict-free timetables within a short time. It automatically detects and resolves scheduling conflicts, ensuring the effective utilization of classrooms, faculty members, and teaching hours. Traditional timetable preparation is a manual and time-consuming process that is prone to human errors, scheduling conflicts, and frequent modifications due to changes in faculty, courses, or classroom availability. These limitations reduce administrative efficiency and create inconvenience for both students and faculty. Therefore, there is a need for an intelligent, automated, and cost-effective timetable generation system that can produce optimized schedules while complying with NEP 2020 guidelines and improving the overall academic planning process.***

KEYWORDS: Artificial Intelligence (AI), Timetable Generation System, National Education Policy (NEP) 2020, Timetable Automation, Optimization Techniques, Conflict-Free Scheduling, Resource Utilization

INTRODUCTION

Educational institutions require efficient timetable management to organize classes, faculty, classrooms, and course schedules. With advancements in Artificial Intelligence (AI), timetable generation can be automated to reduce manual effort and scheduling conflicts. This project presents an AI-Based Timetable Generation System Aligned with NEP 2020 that generates accurate and conflict-free timetables by

considering faculty availability, classroom allocation, and course requirements. The proposed system improves academic planning, ensures better resource utilization, and supports the flexible and student-centered approach of NEP 2020.

LITERATURE SURVEY

Timetable generation has gained significant attention with the advancement of Artificial Intelligence (AI). AI-based timetable systems automatically schedule classes by allocating faculty, classrooms, subjects, and time slots while avoiding scheduling conflicts. Unlike traditional manual methods, these systems improve accuracy, reduce manual effort, and ensure better resource utilization. They also support the flexible and student-centered approach promoted by NEP 2020, making academic planning more efficient and reliable.

RELATED WORK

Existing timetable generation systems use Artificial Intelligence (AI) and optimization techniques to automate scheduling and reduce conflicts. These systems improve resource utilization, minimize manual effort, and generate efficient academic timetables. The proposed system further supports the flexible and student-centered approach of NEP 2020 by creating accurate and optimized schedules.

EXISTING METHOD

The existing timetable generation system mainly depends on manual scheduling by administrators using spreadsheets or traditional software. Timetables are prepared based on faculty availability, classroom allocation, and course schedules, requiring significant time and effort. This process is prone to human errors, scheduling conflicts, and frequent modifications whenever there are changes in faculty, subjects, or classrooms. Traditional systems have limited automation and cannot efficiently handle complex scheduling constraints. Overall, the existing method lacks intelligence, flexibility, and efficient resource utilization.

PROPOSED METHOD

The proposed system implements an AI-Based Timetable Generation System to automate academic scheduling efficiently. It generates conflict-free timetables by considering faculty availability, classroom allocation, course requirements, teaching hours, and institutional constraints. The system uses Artificial Intelligence and optimization techniques to reduce manual effort and scheduling errors. It

automatically updates schedules whenever changes occur, ensuring better resource utilization and efficient academic planning. The proposed system also supports the flexible and student-centered approach of NEP 2020**,** providing a fast, accurate, and reliable solution for educational institutions.

SYSTEM ARCHITECTURE

The AI-Based Timetable Generation System consists of a user interface, AI-based scheduling module, database, and timetable generation module. The administrator enters details such as faculty information, subjects, classrooms, teaching hours, and academic constraints through the user interface. The AI scheduling module processes this information using optimization techniques to generate a conflict-free timetable. All data is stored and managed in the database for easy access and updates.

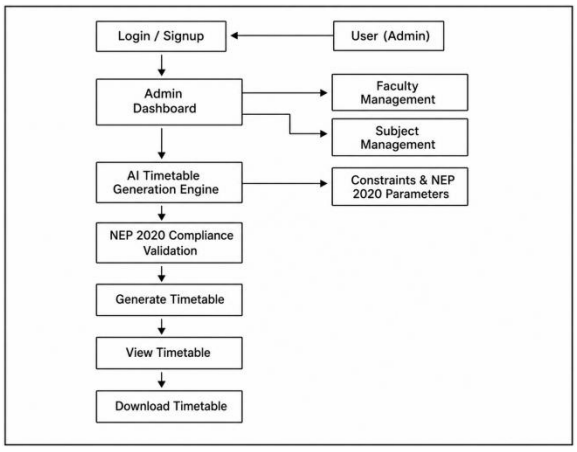

Fig 1: System Architecture

METHODOLOGY DESCRIPTION

The methodology of the AI-Based Timetable Generation System Aligned with NEP 2020 is designed to automatically generate academic timetables using Artificial Intelligence. The system collects information such as faculty details, subjects, classrooms, departments, and available time slots. It processes the data to create a conflict-free and optimized timetable while balancing faculty workload and classroom usage. The system also supports flexible scheduling for theory, practical, and elective courses as per NEP 2020 guidelines. Finally, the generated timetable is stored in the database and made available to administrators, faculty members, and students.

RESULTS AND DISCUSSION

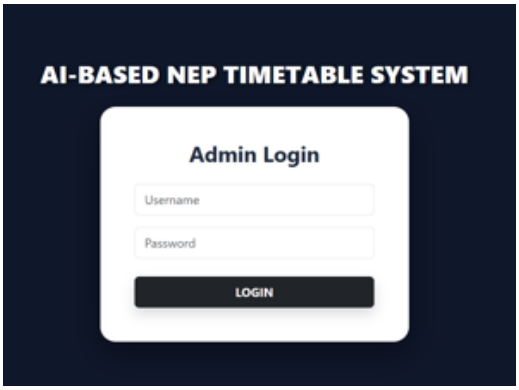

Fig 2: Login Page

The Admin Login Page allows the administrator to securely access the AI-Based Timetable Generation System using

a username and password. After successful login, the administrator can manage all system modules and generate timetables. This ensures secure and authorized access to the system.

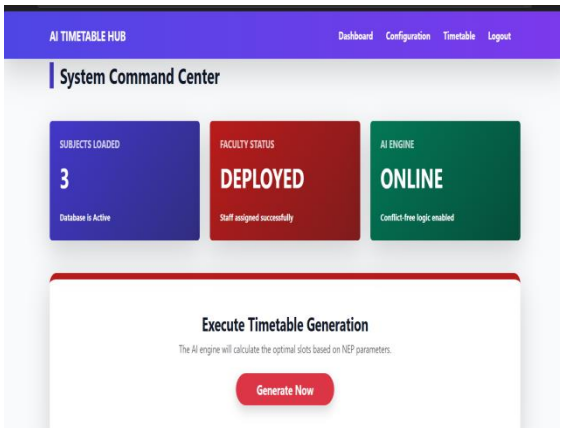

Fig 3: Dashboard Page

The Dashboard provides an overview of the AI-Based Timetable Generation System. It displays important information such as the number of subjects loaded, faculty status, and AI engine status. From this page, the administrator can start the timetable generation process with a single click, making the system simple and efficient.

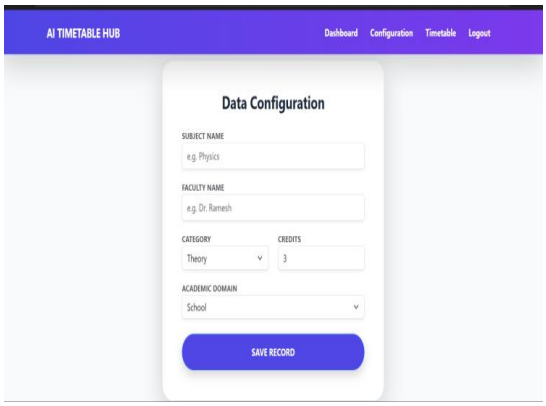

Fig 4: Configuration page

The Data Configuration Page allows the administrator to enter details such as subject name, faculty name, category, credits, and academic domain. The entered data is stored in the database and used by the AI algorithm for timetable generation. This ensures an accurate and conflict-free timetable.

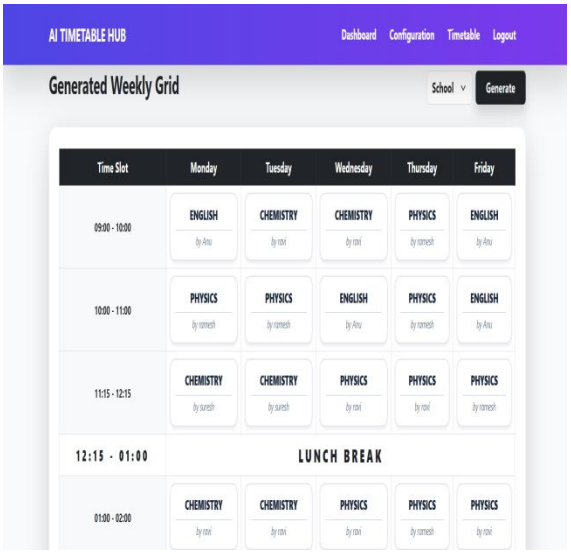

Fig 5: AI-generated Weekly Timetable

The Timetable Generation Page displays the automatically generated weekly timetable in a clear and organized format. The AI algorithm assigns subjects, faculty, and time slots while avoiding scheduling conflicts. This provides an optimized and easy-to-understand timetable for administrators, faculty, and students.

CONCLUSION

The AI-Based Timetable Generation System Aligned with NEP 2020 improves academic scheduling by automatically

generating accurate and conflict-free timetables. By reducing manual effort, balancing faculty workload, and efficiently allocating resources, the system provides a reliable, flexible, and intelligent solution for educational institutions.

FUTURE SCOPE

The proposed AI-Based Timetable Generation System Aligned with NEP 2020 can be further enhanced by integrating advanced Artificial Intelligence and Machine Learning algorithms to generate more accurate and personalized timetables. The system can be extended to support automatic rescheduling in case of faculty leave, classroom unavailability, or sudden timetable changes. Cloud-based storage and mobile applications can be integrated to provide easy access to timetables anytime and anywhere. Support for multiple institutions, departments, and campuses can also be added to improve scalability. Additionally, real-time notifications and AI-based analytics can be incorporated to keep faculty and students informed about timetable updates and optimize academic resource utilization.

REFERENCES

[1] P. Harini, “Sms based wireless e-notice board,” *International Journal of Emerging Technology and Advanced Engineering,* vol. 3, no. 3, pp. 181-185, 2013.

[2] P. Harini, “Enhanced packet delivery techniques using crypto-logic riddle on jamming attacks for wireless communication medium,” *Int. J. Latest Trends Eng. Technol,* vol. 2, no. 4, pp. 469-478, 2013.

[3] P. Harini, “[Retracted] Machine Learning Algorithms Are Applied in Nanomaterial Properties for Nanosecurity,” *Journal of Nanomaterials,* vol. 2022, no. 1, p. 5450826, 2022.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, 4th ed. Cambridge, MA, USA: MIT Press, 2022.

[5] E. K. Burke and P. Ross, The Practice and Theory of Automated Timetabling. Berlin, Germany: Springer, 1996.

[6] E. K. Burke and S. Petrovic, "Recent Research Directions in Automated Timetabling," European Journal of Operational Research, vol. 140, no. 2, pp. 266–280, 2002.

[7] M. W. Carter and G. Laporte, "Recent Developments in Practical Course Timetabling," Lecture Notes in Computer Science, vol. 1153, pp. 3–19, Springer, 1996.

[8] I. Sommerville, Software Engineering, 10th ed. Harlow, U.K.: Pearson Education, 2016.

[9] A. Silberschatz, H. F. Korth, and S. Sudarshan, Database System Concepts, 7th ed. New York, NY, USA: McGraw-Hill Education, 2020.

[10] M. Grinberg, Flask Web Development: Developing Web Applications with Python, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018.

[11] M. Lutz, Learning Python, 5th ed. Sebastopol, CA, USA: O'Reilly Media, 2013.

[12] J. Brownlee, *Machine Learning Mastery with Python*. Melbourne, Australia: Machine Learning Mastery, 2016.

[13] World Wide Web Consortium (W3C), *HTML5: A Vocabulary and Associated APIs for HTML and XHTML*, 2021.

[14] OWASP Foundation, *OWASP Top 10: The Ten Most Critical Web Application Security Risks*, 2021.

[15] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Boston, MA, USA: Addison-Wesley Professional, 1994.