The Pharma Innovation Journal 2019; 8(1): 771-776

www.ThePharmaJournal.com

The Pharma Innovation

ISSN (E): 2277- 7695
ISSN (P): 2349-8242
NAAS Rating: 5.03
TPI 2019; 8(1): 771-776

www.thepharmajournal.com
Received: 09-10-2018
Accepted: 13-11-2018

Neha Sewal
Assistant Professor, Computer Science & Engineering, Lingaya's Vidyapeeth, Faridabad, Haryana, India

Correspondence
Neha Sewal
Assistant Professor, Computer Science & Engineering, Lingaya's Vidyapeeth, Faridabad, Haryana, India

Innovative automated class timetable generator with smart scheduling

Neha Sewal

DOI: https://doi.org/10.22271/tpi.2019.v8.i1m.25412

Abstract
The Automated Class Timetable Generator is a project developed using Node.js, Express, and Mongoose, aimed at efficiently creating class timetables for educational institutions. Timetable generation is a complex task that requires careful consideration of various factors, such as course offerings, faculty availability, classroom capacities, and student preferences. This project leverages the power of Node.js, Express, and Mongoose to automate the process and streamline the creation of class timetables.
The system provides a user-friendly interface where administrators can input the necessary information, such as course details, faculty availability, and classroom capacities. The data is stored in a MongoDB database using Mongoose, allowing for easy retrieval and manipulation. The application incorporates intelligent algorithms to generate optimized timetables that satisfy the constraints and preferences defined by the administrators.
The key features of the Automated Class Timetable Generator include the ability to handle multiple courses, faculty members, and classrooms simultaneously. It ensures that there are no conflicts in scheduling, such as overlapping classes or faculty members assigned to multiple classes at the same time. Additionally, the system considers factors like student preferences for certain time slots or avoiding back-to-back classes.
The project utilizes the asynchronous nature of Node.js and the robustness of Express to handle concurrent user requests and provide real-time updates on timetable generation progress. The generated timetables can be previewed, modified, and exported in various formats for easy distribution to students, faculty, and staff.
Overall, the Automated Class Timetable Generator simplifies the complex task of creating class timetables by leveraging the capabilities of Node.js, Express, and Mongoose. Its user-friendly interface, intelligent algorithms, and real-time updates enhance efficiency and accuracy in timetable generation, benefiting educational institutions by saving time and effort in manual scheduling processes.

Keywords: User friendly, asynchronous nature, real time updates, enhance efficiency, leveraging

Introduction
The process of generating class timetables in educational institutions is often a challenging and time-consuming task. It requires careful consideration of various factors, including course offerings, faculty availability, classroom capacities, and student preferences. Manual timetable creation can be error- prone and inefficient, leading to conflicts, scheduling difficulties, and dissatisfaction among students and faculty. To address these challenges, we present the Automated Class Timetable Generator, a project developed using Node.js, Express, and Mongoose. [Availability, Capacities, error prone, inefficient].
The Automated Class Timetable Generator aims to streamline the process of creating class timetables by automating the scheduling process. By leveraging the power of Node.js, Express, and Mongoose, the system provides an efficient and user-friendly solution for educational institutions to generate optimized timetables that satisfy all necessary constraints and preferences. [streamline, leveraging, constraints].
The project incorporates a web-based interface that allows administrators to input and manage the essential data required for timetable generation. This includes course details, faculty availability, and classroom capacities. The data is stored in a MongoDB database using Mongoose, ensuring efficient retrieval and manipulation. [interface, essential data, retrieval, manipulation] Utilizing intelligent algorithms, the Automated Class Timetable Generator considers various factors to create optimized timetables [1].

It ensures that there are no conflicts in scheduling, such as overlapping classes or faculty members assigned to multiple classes at the same time. Additionally, the system takes into account student preferences, such as preferred time slots or the avoidance of back-to-back classes, to enhance satisfaction and convenience. [enhance, convenience, preferences]

The use of Node.js and Express in the project enables the handling of concurrent user requests and provides real-time updates on the timetable generation progress. This allows administrators to monitor and modify the timetables as needed, ensuring flexibility and adaptability throughout the process. [concurrent, flexibility, adaptability]

The generated timetables can be previewed, modified, and exported in various formats for easy distribution to students, faculty, and staff. This feature enhances communication and transparency within the educational institution, ensuring that all stakeholders have access to up-to-date and accurate class schedules. [stake-holders, accurate, schedule]

In summary, the Automated Class Timetable Generator offers an efficient and reliable solution for educational institutions to automate the complex task of creating class timetables. By leveraging the capabilities of Node.js, Express, and Mongoose, the project simplifies the scheduling process, reduces errors and conflicts, and enhances overall efficiency. The system's user- friendly interface, intelligent algorithms, and real-time updates contribute to improved satisfaction and productivity within educational institutions. [efficient, reliable, complex tasks, algorithms]

Literature Survey

A. Existing System

In the traditional manual approach, the creation of class timetables in educational institutions is a labor-intensive and time-consuming process. Typically, administrators rely on spreadsheets, paper-based forms, and manual coordination to generate timetables. This manual approach is prone to errors, conflicts, and difficulties in accommodating the preferences and constraints of students and faculty members.

In the existing system, administrators often face challenges such as:

Time and Effort: Manual timetable generation requires administrators to spend significant time and effort collecting information, coordinating with faculty members, and manually assigning courses to specific time slots and classrooms.

Error-Prone: Human errors, such as scheduling conflicts, overlapping classes, or double bookings of classrooms or faculty, can occur during the manual timetable creation process. These errors can lead to confusion, frustration, and disruption of the academic schedule.

Limited Flexibility: Making changes or modifications to the timetable becomes a cumbersome task in the manual system. Adjusting schedules to accommodate unforeseen circumstances or changing requirements can be challenging and time- consuming [2].

Lack of Optimization: The manual approach may not consider optimization factors such as faculty preferences, student preferences, or classroom capacities. This can result in inefficient use of resources and dissatisfaction among stakeholders.

Communication and Distribution: Communicating and distributing the timetables to students, faculty, and staff can be challenging in a manual system. It may involve printing physical copies or sharing spreadsheets, which can lead to confusion and difficulties in accessing up-to-date information.

Due to these limitations, the existing manual system for generating class timetables often causes inefficiencies, errors, and dissatisfaction among stakeholders. Therefore, there is a need for an automated solution that overcomes these challenges and streamlines the process of creating class timetables in educational institutions [3].

Problems in Existing System

The existing manual system for generating class timetables in educational institutions suffers from several limitations and problems, including:

Time-Consuming Process: The manual creation of timetables is a time-consuming task that requires administrators to collect information from various sources, coordinate with faculty members, and manually assign courses to specific time slots and classrooms. This process can take a significant amount of time, especially for institutions with a large number of courses and faculty members.

Error-Prone: Manual timetable creation is prone to errors, such as scheduling conflicts, overlapping classes, or double bookings of classrooms or faculty members. These errors can lead to confusion, disruptions in the academic schedule, and dissatisfaction among students and faculty.

Limited Optimization: The manual system often lacks the ability to optimize timetables based on various factors such as faculty preferences, student preferences, classroom capacities, and other constraints. This can result in inefficient use of resources, such as classrooms and faculty availability, leading to suboptimal timetables.

Lack of Flexibility: Making changes or modifications to the timetable in the manual system can be a cumbersome and time- consuming process. Adjusting schedules to accommodate unforeseen circumstances or changing requirements requires extensive manual coordination and adjustments, leading to inefficiencies and delays.

Communication and Distribution Challenges: Sharing and distributing the timetables to students, faculty, and staff in the manual system can be challenging. It often involves printing physical copies or sharing spreadsheets, which can lead to difficulties in accessing up-to-date information and may result in miscommunication or confusion regarding schedule changes.

Difficulty in Meeting Preferences: The manual system may struggle to accommodate individual preferences and constraints of students and faculty members. It may be challenging to take into account specific time preferences, course preferences, or avoid back-to-back classes, resulting in reduced satisfaction and inconvenience.

Scalability Issues: As the number of courses, faculty members, and students increases, managing the manual timetable creation process becomes increasingly complex. It

becomes difficult for administrators to handle the growing workload efficiently and maintain accuracy in the schedules. These problems highlight the need for an automated solution that can streamline the timetable generation process, optimize schedules, accommodate preferences and constraints, and improve efficiency and accuracy in educational institutions.

Proposed System

To overcome the limitations and challenges of the existing manual system, we propose the development of an Automated Class Timetable Generator using Node.js, Express, and Mongoose. The proposed system aims to automate and streamline the process of creating class timetables in educational institutions, providing an efficient and optimized solution.

The key features and functionalities of the proposed system include

User-Friendly Interface: The system will provide a user-friendly web-based interface for administrators to input and manage all the necessary data required for timetable generation. This includes course details, faculty availability, classroom capacities, and student preferences.

Data Management: The system will utilize MongoDB as a database and employ Mongoose as an Object Data Modeling (ODM) tool to store and manage the data. This allows for efficient retrieval, manipulation, and storage of information related to courses, faculty members, classrooms, and other scheduling parameters.

Automated Timetable Generation: The system will incorporate intelligent algorithms to automatically generate optimized timetables. These algorithms will consider various factors such as course offerings, faculty availability, classroom capacities, student preferences, and other constraints. The goal is to create conflict-free timetables that maximize resource utilization and meet the preferences of students and faculty.

Real-time Updates and Notifications

The system will leverage the asynchronous nature of Node.js and the robustness of Express to handle concurrent user requests and provide real-time updates on the timetable generation progress. Administrators will be able to monitor the process, track any conflicts or issues, and receive notifications when the timetables are generated or modified.

Flexibility and Customization

The proposed system will offer flexibility and customization options to accommodate changes, adjustments, and unforeseen circumstances. Administrators will be able to make modifications to the generated timetables, such as resolving conflicts, accommodating specific preferences, or adjusting schedules as needed.

Export and Distribution

The system will allow administrators to preview, modify, and export the generated timetables in various formats, such as PDF or Excel. This enables easy distribution of the timetables to students, faculty, and staff, ensuring everyone has access to up-to-date and accurate class schedules.

Scalability and Performance: The proposed system will be designed to handle a large number of courses, faculty members, and students, ensuring scalability and optimal performance. It will efficiently manage the increasing workload and maintain accuracy even as the educational institution grows. By implementing the proposed Automated Class Timetable Generator, educational institutions can benefit from reduced time and effort in timetable creation, minimized errors and conflicts, improved resource utilization, enhanced satisfaction among students and faculty, and streamlined communication and distribution of schedules.

Literature Survey: The {1} is an Automatic Timetable generator. This paper is based on converting a manual system of arranging the time table into an automated way using the tools like Python, PyCharm, XAMPP, MYSQL, DJANGO. The source from which this
paper was taken is IRJET. The Evaluation parameter is Creating a proper Timetable.

The {2} is an Automatic Timetable generator. Carter and Laporte considered different categories to solve the timetabling problem. They are cluster method, Meta-Heuristics and Constraint-Based method. Some of the Meta Heuristics methods are TABU which was found to outperform all other methods. But its complexity makes it very difficult to use it in a course timetabling problem in a short period of time. Hence, they have decided to take the best aspects of GA and TABU and came up with an application-oriented algorithm designed specifically for the needs of our college. The source is from JETIR.

The {3} is an Automatic Timetable generator. This paper is based on using a timetable object for customizing the algorithm. This object comprises classroom objects and the timetable for them likewise a fitness score for the timetable. Tools and the software used are Microsoft .Net Framework, Microsoft SQL Server, Middleware Technology. The source from which this paper is taken is IJSR. The Evaluation parameter is Creating a proper Timetable.

The {4} is an Automatic Timetable generator. This paper is based on converting a manual system of arranging the time table into an easy way using the tools of Java-based software. The source from which this paper is taken is the International Journal of Advanced Research in Computer Science Software Engineering (IJAECSSE). The Evaluation parameter is creating a proper Timetable [4].

Table 1: About The research papers

Article/Author	Year and Citation	Tools/Software	Source	Evaluation Parameter
Prof. Pravin Patil	2021	Python, Pycharm, XAMPP, Mysql, DJANGO	IRJET (International Research Journal of Engineering and Technology	One way for generating a time Table
Deeksha C S	2015	TABU and GA	JETIR	One way for generating a time Table
Yash Lahoti	2017	Microsoft	IJSR	One way for generating a time Table

		.Net Framework, Microsoft SQL Server, Middle ware Technology		
Saritha M	2017	Java based software	IJARCSSE	One way for generating a time Table

Hardware and Software Requirements

Hardware Requirements

1. The system wants a minimum of two GB of ram to run all the options. It wants a minimum 1.3 GHz processor to run smoothly. Rest is all up to the user's usage and can take care of hardware. For security opposing anti-virus is suggested.
2. RAM: At least 256 MB of RAM. The amount of RAM needed depends on the number of concurrent client connections, and whether the server and multiplexor are deployed on the same host.
3. Disk Space: Approximately 300 MB required for Instant Messaging Server software. Processor: Minimum 1.3 gigahertz (GHz) x86- or x64-bit dual core processor with SSE2instruction set and recommended 3.3 gigahertz (GHz) or faster 64-bit dual core processor with SSE2 instruction set.
4. Memory: Minimum 2-GB RAM and recommended 4-GB RAM or more.

Software Requirements

The functions performed by the computer are the result of the software, the computer is capable of performing many different functions. Some basic initial software must be installed on a computer for an application to function optimally.

- VS Code
- Web Browser
- Node JS
- Express
- Mongoose

Feature / Characteristics Identification

Here are some possible features/characteristics of an Automatic

Class Time Table generator

Inputting proper data: The system should be able to take input data such as add the course name, Teacher to be taught, number of periods.

Schedule proper generation: The Proposed system will be able to generate a schedule for classes based on the input data that is entered in Add new course and generate time table. The schedule should be optimized for factors such as the availability of teachers and classrooms, class sizes, and other constraints.

Customizable preferences: Here the user can customize the preferences by going to update time tables, can also delete the time table as they like to customize it.

User-friendly interface: The proposed system will have an intuitive and easy-to-use interface that allows users to input data, view the time-table, and make adjustments or customize as needed.

Scalability: The proposed system can be scalable and able to handle large amounts of data as the number of students, teachers, and classrooms increases over time and can help in scheduling a well-structured time table.

Flexibility: The proposed system will be flexible enough to accommodate changes in the schedule by going to update time table and update database, such as adding or removing classes, or adjusting class timings.

Security: The proposed system will be designed with appropriate security measures to ensure the confidentiality of the user that only the user can login to his account an nobody else and integrity of the input data.

Constraint Identification

1. Sometimes the workload of a staff may be more than the prescribed.
2. Taking clarity of the subject hours and the other hours for each class.
3. Introducing flexibility of number of hours per day and to generate the corresponding timetables.
4. Student preferences: Students might have preferences for certain subjects or may need to attend classes at specific times due to other commitments. The proposed system should take into account student preferences and needs while generating the class schedule when the user clicks on generate time table.
5. Class size: The size of the class can affect the availability of resources, and the system needs to ensure that the class sizes are balanced across different subjects.
6. Overlapping classes: The system needs to ensure that there are no overlapping classes, where students are required to attend two classes at the same time.

Analysis of Features and Finalization Subject to Constraints After identifying the features and constraints of an Automatic Class Time Table Generator, it is important to analyze the feasibility of the proposed features and finalize the system subject to the identified constraints.

Inputting proper data: The system should be able to take input data such as add the course name, Teacher to be taught, number of periods. This feature is feasible and can be included in the system subject to the constraints of resource availability, room allocation, and class size.

Schedule proper generation: The Proposed system will be able to generate a schedule for classes based on the input data that is entered in Add new course and generate time table. This feature is feasible but subject to constraints such as time availability, teacher workload.

Customizable preferences: Here the user can customize the preferences by going to update time tables, can also delete the time table as they like to customize it. This feature is feasible but subject to constraints such as resource availability, and teacher workload.

User-friendly interface: The proposed system will have an

intuitive and easy-to-use interface that allows users to input data, view the time-table, and make adjustments or customize as needed. This feature is feasible and will be included in the system.

Scalability: The proposed system can be scalable and able to handle large amounts of data as the number of students, teachers, and classrooms increases over time and can help in scheduling a well-structured time table. This feature is feasible but subject to constraints such as resource availability and time constraints.

Flexibility: The proposed system will be flexible enough to accommodate changes in the schedule by going to update time table and update database, such as adding or removing classes, or adjusting class timings. This feature is feasible but subject to constraints such as resource availability.

Security: The proposed system will be designed with appropriate security measures to ensure the confidentiality of the user that only the user can login to his account an nobody else and integrity of the input data. This feature is feasible and will be included in the system.

Design Selection

Design selection for an automated class time table generator using JavaScript, Express, and Mongoose involves several considerations:

User needs: The design should meet the needs of the users, which may include instructors, students, and administrators. The design should be intuitive and easy to use, and should support the specific requirements of each user group.

Functionality: The design should support the required functionality of the application, such as the ability to create and manage schedules, assign instructors to classes, and manage student enrolments.

Scalability: The design should be scalable, allowing the application to grow and support additional classes, instructors, and students as needed.

Security: The design should incorporate appropriate security measures to protect sensitive information such as student records, instructor schedules, and class assignments.

Performance: The design should be optimized for performance, minimizing load times and ensuring that the application can handle large amounts of data and user traffic.

Maintainability: The design should be maintainable, allowing for easy updates and modifications as needed.

Integration: The design should be compatible with other systems and applications used within the organization, such as student information systems, learning management systems, and others.

Ultimately, the design selected for an automated class time table generator should be based on a careful analysis of the user requirements, functional requirements, and technical considerations. It may be helpful to conduct user testing and gather feedback from stakeholders to ensure that the design meets their needs and expectations.

Methodologies

Requirements gathering: Identify the requirements of the educational institution, such as the number of students, courses, instructors, and rooms, and the constraints and preferences for scheduling.

Data modelling: Define the data schema and models using Mongoose, which will be used to store and retrieve data from the MongoDB database.

Server-side development: Develop the server-side logic using Node.js and Express, which will handle the HTTP requests and responses and communicate with the database.

Client-side development: Develop the client-side user interface using HTML, CSS, and JavaScript, which will allow users to view the class time table, request changes, and communicate with each other.

Integration and testing: Integrate the server-side and client-side components and test the application to ensure that it meets the requirements and functions correctly.

Deployment and maintenance: Deploy the application to a production environment and maintain it to ensure that it remains up-to-date and secure.

In terms of the specific technologies used, JavaScript is used for both server-side and client-side development, while Express is used for server-side development and Mongoose is used for data modelling and interaction with MongoDB. The application is typically developed using an Agile methodology, which emphasizes collaboration, flexibility, and iterative development.

Overall, the methodology used in an automated class time table generator using JavaScript, Express, and Mongoose focuses on gathering requirements, modelling data, developing server-side and client-side components, testing, and deployment. It emphasizes collaboration, flexibility, and iterative development to ensure that the application meets the needs of the educational institution and its users.

Result/Output

The result of implementing an automated class time table generator using Node.js, Express, and Mongoose would be a robust and efficient solution for creating and managing class schedules in educational institutions. Here are some key outcomes of the project:

Streamlined Scheduling Process: The automated class time table generator simplifies the scheduling process by automating the creation of class schedules. It eliminates the need for manual scheduling, reducing errors and saving time for administrators and staff.

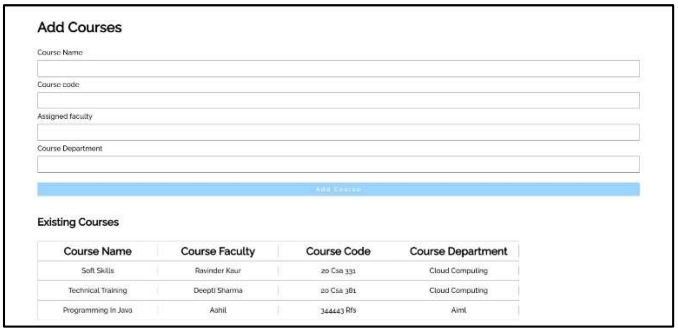
Improved Efficiency: The use of Node.js, Express, and Mongoose enables the application to handle a large number of simultaneous connections efficiently. This allows for fast and responsive performance, ensuring smooth scheduling operations even with multiple users accessing the system simultaneously.

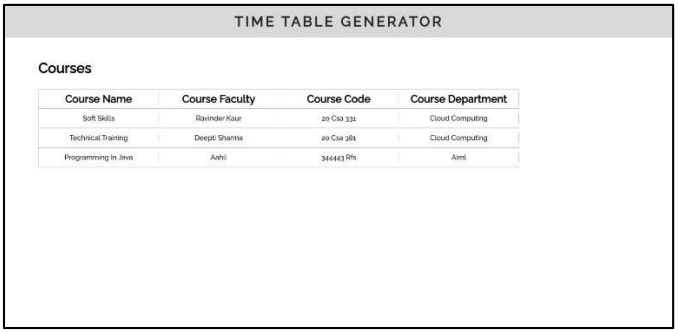
Real-time Updates and Notifications: The application can provide real-time updates and notifications for changes in the class schedule. This enhances communication between

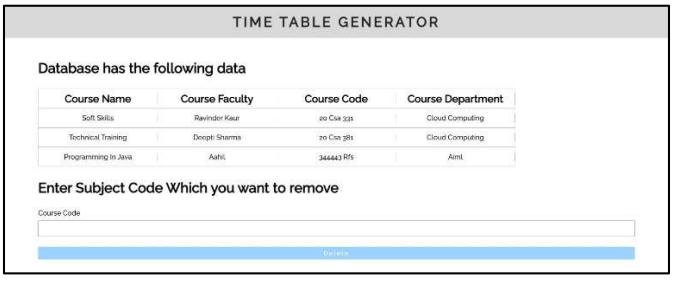
students, teachers, and administrators, keeping them informed about any modifications or cancellations.

User-Friendly Interface: The front-end interface developed using HTML, CSS, and JavaScript provides a user-friendly experience. It can include features such as search, filter, and sort options, making it easy for users to navigate and access the class schedules.

Implementation







Conclusion

The Automated Class Timetable Generator developed using Node.js, Express, and Mongoose offers a significant advancement over the traditional manual approach to class timetable generation in educational institutions. By automating the process and incorporating intelligent algorithms, the proposed system addresses the limitations and challenges of the existing system.

The project provides a user-friendly interface for administrators to input and manage essential data, such as course details, faculty availability, and classroom capacities. Leveraging the power of Node.js, Express, and Mongoose, the system efficiently handles the data, ensuring easy retrieval, manipulation, and storage.

With its automated timetable generation capabilities, the system optimizes the scheduling process by considering various factors, including constraints and preferences of students and faculty. It eliminates scheduling conflicts, maximizes resource utilization, and enhances overall satisfaction and convenience.

Real-time updates and notifications enable administrators to monitor the progress of timetable generation and make necessary modifications. The system offers flexibility and customization options, allowing adjustments to accommodate unforeseen circumstances and changing requirements.

The ability to export timetables in different formats facilitates easy distribution to students, faculty, and staff, improving communication and accessibility to up-to-date class schedules. Additionally, the scalability and performance of the system ensure its effectiveness even as the institution grows.

Overall, the Automated Class Timetable Generator streamlines the timetable generation process, saves time and effort, minimizes errors and conflicts, and enhances resource utilization and stakeholder satisfaction. By embracing automation and intelligent algorithms, educational institutions can optimize their scheduling processes, improve productivity, and enhance the overall academic experience for students and faculty.

References

1. Automatic Timetable Generator [Internet]. [cited 2022 Jan 1]. Available from: [URL]
2. Chanana N, Goele S. Future of e-commerce in India. Int J Comput Bus Res. 2012;8.
3. IEEE. Dyl T, Przeorski K. Mastering Full-Stack React Web Development. Packt Publishing; 2017. Ambler T, Cloud N. Javascript Frameworks for Modern Web Dev. Apress; 2015.
4. Kaushik P, Yadav R. Reliability design protocol and blockchain locating technique for mobile agent. J Adv Sci Technol (JAST). 2017;14(1):136-141. Available from: https://doi.org/10.29070/JAST