IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

## IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 5, May 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205296 |

# Scheduloom: Scheduling Software

**Kush Patel, Kasturi Patil, Harshvardhan Khoje, Chetna Takne, Dipika. L. Tidke**

UG Student, Department of Computer Engineering, MVPS's KBTCOE Nashik, India

Assistant Professor, Department of Computer Engineering, MVPS's KBTCOE Nashik, India

**ABSTRACT**: The scheduling system for a computer engineering department orchestrates comprehensive timetables encompassing multiple academic years. Utilizing web based technologies, including HTML, CSS, JavaScript, Angular, and Node.js, the system harmonizes intricate inputs such as lecture details, staff allocations, classroom assignments, and practical session scheduling. Employing an intuitive user interface, it leverages a sophisticated scheduling algorithm to optimize resource utilization, ensuring faculty availability, and accommodating student preferences. Through real-time updates and interactive functionalities, it delivers a streamlined, adaptable scheduling solution tailored to the dynamic needs of a diverse academic curriculum. This abstract encapsulates the essence of the system, highlighting its technological framework, the complexity of inputs managed, and its ability to cater to the dynamic requirements of a computer engineering department's scheduling needs.

**KEYWORDS**: Timetable; Heuristic Approach; Scheduling; Constraint; Optimization

## I.INTRODUCTION

Designing an effective scheduling system for a computer engineering department necessitates a comprehensive understanding of the intricate web of academic timetables, faculty constraints, and student dynamics. In the contemporary landscape of educational management, the amalgamation of advanced technologies, dynamic user interfaces, and sophisticated algorithms forms the bedrock of a scheduling solution capable of addressing multifaceted challenges. At the heart of this scheduling endeavor lies the integration of HTML, CSS, JavaScript, Angular, and Node.js, converging to build a robust and interactive web-based system. This amalgamation propels the scheduling process beyond mere data management, offering a dynamic platform that seamlessly interfaces with stakeholders across the academic spectrum. Harnessing the versatility of HTML and CSS, the system crafts an intuitive and visually appealing interface, rendering user interactions smooth and navigable.

Within the purview of the computer engineering department's scheduling demands lies a multifaceted labyrinth of inputs and considerations. Managing lecture details, staff allocations, classroom assignments, and practical session schedules necessitates an intricate orchestration of data and constraints[4]. The system encapsulates these complexities, providing an intuitive interface for administrators and coordinators to input and manage a myriad of details effortlessly. Its sophisticated scheduling algorithm serves as the nucleus of the system, seamlessly maneuvering through diverse inputs to generate optimized timetables. Balancing faculty availability, lecture hours, practical session allocations, and classroom constraints, the algorithm operates within the confines of preset rules and dynamic adjustments, optimizing resource utilization while ensuring minimal conflicts.

Furthermore, the system's adaptability and responsiveness manifest through real-time updates and interactive functionalities[5]. Users can visualize and manipulate timetables, receiving immediate feedback on changes, thereby fostering an environment of collaboration and transparency. Additionally, automated notifications ensure stakeholders remain informed of schedule modifications, promoting an ecosystem of seamless communication and enhanced productivity.

## II.RELATED WORK

In their work on automating a timetable generator, yash lahoti, aaditya punekar, hiten patel, and vishal bhimsariya (2021) highlight the challenges associated with creating timetables. They identify a significant obstacle in the form of numerous restrictions, which include environmental fluctuations. These constraints add complexity to the timetable generation process, making it a challenging task. Addressing these issues and automating the generation of timetables is crucial, as it streamlines the process and helps ensure efficiency in scheduling, especially in dynamic environments where external factors may impact the scheduling requirements[1].in their endeavor to create a sophisticated timetable system, mujahid tabassum and norita md norwani (2019) focus on the development of an efficient solution using

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205296 |

angularjs and bootstrap. The software they propose aims to provide users with a professional graphical user interface (gui) that not only enhances user experience but also significantly saves time. By incorporating angularjs, a powerful javascript framework, and bootstrap, a front-end framework known for its responsiveness and sleek design, the researchers ensure a visually appealing and flexible interface[2].in their research titled "intelligent timetable scheduler: a comparison of genetic, graph coloring, heuristic, and iterated local search algorithms," conducted in 2018, tiny wijerathna ekanayake, shawn ragel, pavani subasinghe, anjalie gamage, and suchini attanayaka delve into the realm of intelligent timetable scheduling. The study involves a comprehensive exploration and evaluation of four distinct algorithms: genetic, graph coloring, heuristic, and iterated local search[3].

## III.PROPOSED ALGORITHM

**Input** Number of Lectures per week, Staff and
Faculty Details, Classroom Number, Subject Details,
Practical Details.
**Output:** Timetable satisfying all the constraints.
**Software:**
**Accepts** Inputs
**Creates** Database
**Checks** Constraints
**Maps** Schedule
**Displays** Output

## IV. OBJECTIVES

● **Optimization of Resources:** Ensure efficient utilization of resources such as classrooms, teachers, and equipment to minimize conflicts and maximize availability.
● **Minimization of Time Gaps:** Minimize time gaps between classes for students and teachers to improve productivity and utilize time effectively.
● **Balanced Workload**: Distribute workload evenly among teachers by assigning a balanced number of classes and diverse subjects.
● **Avoidance of Conflicts:** Prevent clashes between classes, exams, or other events to maintain smooth operation and avoid confusion.
● **Accommodation of Preferences:** Incorporate preferences and constraints provided by users, such as preferred class timings, teacher preferences, or specific room requirements.
● **Handling of Constraints:** Manage various constraints such as teacher availability, room capacity, and student preferences while generating the timetable.
● **Adaptability to Changes:** Create a flexible timetable that can accommodate unexpected changes or adjustments without disrupting the entire schedule.
● **Scalability:** Design the algorithm to handle large datasets efficiently, ensuring scalability as the size of the institution or schedule complexity increases.
● **User-Friendly Interface:** Develop an intuitive and user-friendly interface for inputting constraints, viewing generated adjustments as needed. timetables, and making
● **Performance Optimization:** Optimize the heuristic algorithm for speed and accuracy to generate timetables within a reasonable time frame while maintaining quality.
● **Evaluation Metrics:** Provide metrics for evaluating the quality of generated timetables, such as overall conflicts, time gap distribution, and workload balance.
● **Integration:** Allow integration with existing academic management systems or tools to streamline the scheduling process and facilitate data exchange

## V.ADVANTAGES

1. Time Efficiency: Scheduling software automates the often time-consuming process of creating and managing schedules. It can generate optimized timetables quickly, reducing administrative workload and human errors.
2. Resource Optimization: The software can efficiently allocate resources, such as teachers, classrooms, and equipment, resulting in better resource utilization, cost savings, and improved productivity.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205296 |

3. Flexibility and Adaptability: Scheduling software can adapt to a wide range of scheduling scenarios, from educational institutions with diverse courses to businesses managing shifts or appointments. It offers flexibility to accommodate changing needs
4. Accuracy and Consistency: By relying on algorithms and predefined rules, scheduling software ensures consistent and error-free scheduling. This accuracy leads to fewer scheduling conflicts and improved overall quality
5. Data Analysis and Reporting: Scheduling software provides valuable insights into resource usage and scheduling efficiency. It enables data analysis and reporting, helping organizations make informed decisions and refine their scheduling processes.

## VI. APPLICATIONS

1. **Educational Institutions**: Educational institutions, from schools to universities, use scheduling software to create and manage class schedules. It optimizes teacherstudent allocations, classroom assignments, and resource utilization.
2. **Healthcare Facilities**: Hospitals, clinics, and healthcare centers employ scheduling software to manage appointments, surgeries, and staff shifts. It helps reduce patient wait times and improve healthcare service efficiency.
3. **Businesses and Workplaces**: Businesses use scheduling software for employee shift planning, project management, and meeting scheduling. It optimizes workforce allocation and resource usage.
4. **Transportation and Logistics**: In the transportation and logistics industry, scheduling software is used for route optimization, vehicle allocation, and delivery scheduling. It ensures efficient movement of goods and services.
5. **Event Management**: Event planners and organizers rely on scheduling software to coordinate events, conferences, and meetings. It helps manage venues, speakers, and other event resources effectively.

## VII. METHODOLOGY AND DISCUSSION

The methodology for creating the scheduling software for the computer engineering department revolves around an iterative and collaborative approach, encompassing several interconnected stages of development, testing, and refinement. Initiating the process involves an in-depth analysis and understanding of the department's scheduling intricacies, encompassing the examination of course structures, faculty constraints, and classroom resources[6]. This phase, characterized by extensive stakeholder consultations and requirement gathering, lays the groundwork for the subsequent phases.

Subsequently, the system design phase commences, where architectural blueprints and wireframes are crafted, mapping out the user interface, system modules, and database architecture. Simultaneously, the technological stack, incorporating HTML, CSS, JavaScript, Angular, and Node.js, is strategically organized to underpin the system's robustness and scalability.

The development phase unfolds through an agile methodology, fostering iterative cycles of implementation, feedback, and enhancement. The front-end interface, leveraging HTML, CSS, and JavaScript frameworks, takes shape through responsive design and interactive elements, while the back-end architecture, powered by Node.js, orchestrates data management and computational algorithms[7].

Parallelly, the core of the system—the scheduling algorithm—is fine-tuned and integrated. This algorithm, engineered to navigate the complexities of lecture details, staff allocations, practical sessions, and classroom constraints, undergoes rigorous testing and optimization to ensure efficiency and accuracy. The iterative nature of the methodology emphasizes constant user feedback and testing. As the system evolves, stakeholders are actively engaged, providing insights and refinements that inform the system's ongoing development[8]. User acceptance testing and quality assurance processes are integral, ensuring the system meets the requirements and functions seamlessly across diverse scenarios.

Additionally, the methodology prioritizes scalability and future adaptability. The architecture and codebase are structured to accommodate future expansions, evolving course structures, and technological advancements, thus future-proofing the scheduling solution. Throughout the entire methodology, collaboration among multidisciplinary teams—developers, designers, stakeholders, and end-users—is central. Regular communication, feedback loops, and incremental improvements foster an environment of continuous improvement, culminating in the creation of a scheduling software that aligns closely with the department's dynamic and evolving needs.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205296 |

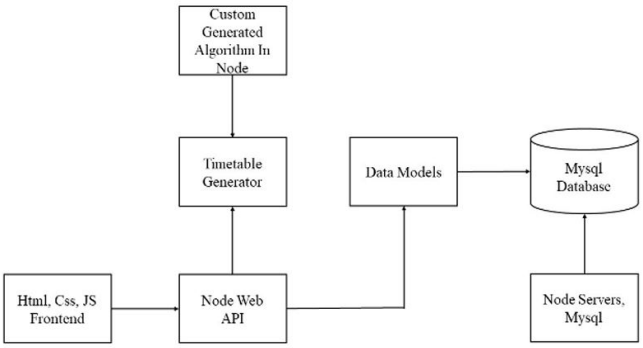


Figure 1. Architecture diagram

## VIII. DATA FLOW OF THE SYSTEM

The data flow within the scheduling system for the computer engineering department encompasses a seamless exchange of information across various stages, modules, and user interactions. At its core, the system ingests inputs such as course details, faculty availability, lecture hours, practical session requirements, and classroom resources. These inputs serve as the foundational data elements, feeding into the scheduling algorithm and the system's database[9]. The scheduling algorithm processes this data, applying rules, constraints, and optimization criteria to generate comprehensive timetables for multiple academic years. Simultaneously, the front-end user interface enables administrators and coordinators to input, manipulate, and visualize scheduling data interactively, facilitating real-time adjustments and updates. The output generated by the scheduling algorithm, comprising finalized timetables, is relayed back to the database, ensuring consistency and accessibility across the system. Additionally, the system's notifications and reporting mechanisms facilitate the dissemination of schedule changes and updates to stakeholders, fostering a dynamic and informed environment. This continuous flow of data, from inputs to algorithmic processing and user interactions, forms the backbone of the scheduling system's functionality, ensuring accuracy, responsiveness, and adaptability to the department's scheduling needs.

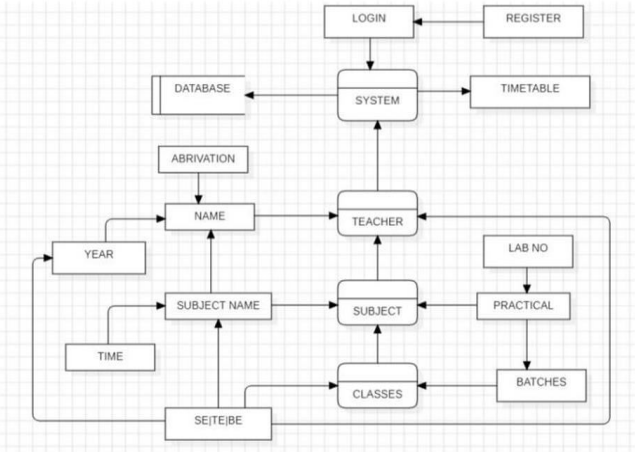


FIGURE 2. DATA FLOW DIAGRAM

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205296 |

## IX. RESULTS

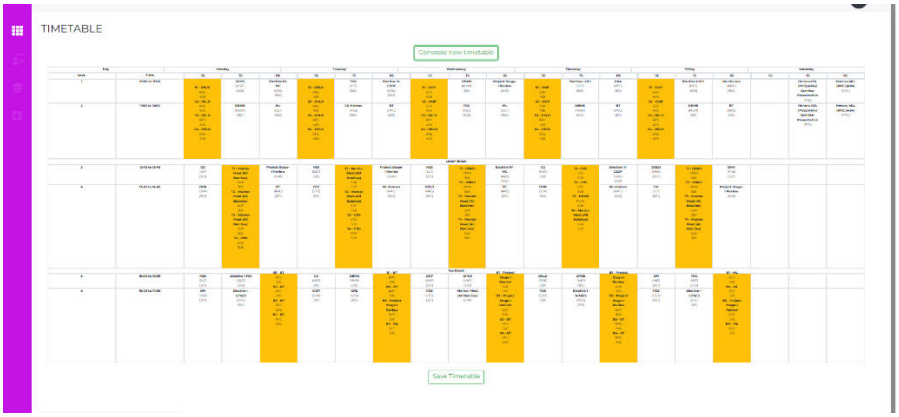


Fig.3. Genrated Timetable

## X.CONCLUSION AND FUTURE WORK

In conclusion, the development of a scheduling software solution is poised to transform the management of timetables for educational institutions, businesses, and organizations. This project's focus on advanced scheduling algorithms, user customization, and compliance with accessibility and security standards reflects a commitment to efficiency, adaptability, and inclusivity. The successful execution of this project hinges on the effective management of technical, operational, and regulatory risks. Proactive risk mitigation, monitoring, and management strategies are vital to ensuring a smooth project journey. Additionally, assumptions and dependencies play a pivotal role in project progress and must be continuously monitored and addressed. By adhering to these principles and requirements, the scheduling software project is poised to streamline scheduling processes, improve resource allocation, and enhance. user satisfaction. Its potential to adapt to diverse scheduling scenarios, integrate with external systems, and deliver secure, accurate schedules positions it as a valuable asset in various sectors. With the right blend of technology, robust risk management, and user-focused design, this project holds the promise of creating a scheduling solution that brings efficiency and effectiveness to an array of scheduling challenges.

## REFERENCES

[1] Yadav, S. (2021). College Time Table Generation. Scribd.
[2] Gupta, A., et al. (2011). Automated College Time Table Generation System. IEEE Xplore.
[3] Verma, R., et al. (2020). A Hybrid Approach for College Timetable Generation. IEEE Xplore.
[4] Kuemar, P. (2018). Time Table Generation. International Journal of Innovative Science and Research Technology.
[5] Sharma, S., et al. (2021). An Intelligent Approach to Automated College Timetable Generation. IEEE Xplore.
[6] Singh, M., et al. (2021). Optimization of College Timetable Generation Using Genetic Algorithm. IEEE Xplore.
[7] Lahoti, Y., Punekar, A., Patel, H., Bhimsariya, V., et al. (2021). Automate Timetable Generator.
[8] Tabassum, M., Norwani, N. M. D. (2019). Development of an Efficient Timetable System using AngularJs and Bootstrap.
[9] Ekanayake, T. W., Ragel, S., Subasinghe, P., Gamage, A., Attanayaka, S. (2018). Intelligent Timetable Scheduler: A comparison of Genetic, Graph Coloring, Heuristic and Iterated Local Search Algorithms.
[10] Fatt, A.C.M., Kee, C.W., Heong, L.C., Seng, N.H., Har, K.N.S. (2015). Software Engineering Approach for a Timetable Generator.
[11] Jain, N., Agarwal, V., et al. (2018). A Survey on Timetable Generation Techniques for Educational Institutions. International Journal of Computer Applications, 181(36).
[12] Kumar, R., Gupta, S. (2018). Design and Implementation of Automated College Timetable Generation System Using Constraint Satisfaction Problem. International Applications, 180(35). Journal of Computer
[13] Patel, A., Shah, B. (2018). Efficient Timetable Generation Using Machine Learning Algorithms. International Journal of Computer Applications, 179(26).
[14] Reddy, R., Kumar, S. (2018). Fuzzy Logic Based Approach for College Timetable Generation. International Journal of Computer Applications, 178(24).
[15] Desai, T., Mehta, S. (2018). Hybrid Approach for College Timetable Generation Using Genetic Algorithm and Simulated Annealing. International Journal of Computer Applications, 177(8).

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

## IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details