ISSN 2277-2685
IJESR/April-June. 2025/ Vol-15/Issue-2s/635-644
Kota Keerthi *et. al.*, / International Journal of Engineering & Science Research

Schedule Sync

[1]Mr Mohammed Arshad Hussain, [2]Kota Keerthi, [3]Punna Pallavi

[1]Assistant professor, Department of CSE, Bhoj Reddy Engineering College for Women, India

[2,3]B.Tech Students, Department of CSE, Bhoj Reddy Engineering College for Women, India

ABSTRACT

Timetabling concerns all activities with regard to producing a schedule that must be subjective to different constraints. Timetable can be defined as the optimization of given activities, actions, or events to a set of objects in space-time matrix to satisfy a set of desirable constraints. A college timetable is a temporal arrangement of a set of lectures and classrooms in which all given constraints are satisfied. Creating such timetables manually is a complex and time-consuming process. By automating this process with a computer-assisted timetable generator, we can save a lot of precious time for administrators who are involved in creating and managing course timetables. Since every college has its own timetabling problem, the commercially available software packages may not suit the needs of every college. Hence, we have developed a practical approach for building lecture course timetabling systems, which can be customized to fit any college's timetabling problem. This project introduces a practical timetabling algorithm capable of taking care of both strong and weak constraints effectively, used in an automated timetabling system. So that each teacher and student can view their timetable once they are finalized for a given semester, but they can't edit them.

1. INTRODUCTION

In today's fast-paced world, efficient time management is crucial for individuals and organizations alike. One of the fundamental tools for managing time effectively is a well-structured timetable. However, creating a timetable manually can be a tedious and time-consuming task, especially when dealing with multiple constraints and variables. To address this challenge, the development of a Timetable Generator Application has been undertaken as a mini project.

Existing System

Generally, timetable generation is done manually. Present day preparator of timetable use MS EXCEL and MS WORD. This way of preparing timetable does not make the system efficient and makes it look more complex it is also a time consuming process. The process of timetable generation, when done manually, often involves significant effort and consumes considerable time. In many institutions, staff members use tools like Microsoft Excel and Microsoft Word to prepare timetables. While these tools provide basic organizational features, they are not optimized for the complexities of timetable management.

Proposed System

The Timetable Generator Application will offer a comprehensive set of features designed to simplify and streamline the timetable creation and management process. It will include user-friendly input forms for collecting essential data, such as class schedules, resource availability, and constraints, ensuring an intuitive data entry experience. The application will utilize advanced algorithms to automate the scheduling process, generating optimized timetables that meet institutional requirements and user preferences.

A modern, user-friendly interface that simplifies data input, navigation, and timetable adjustments.

2. Literature survey

1. D. Nguyen, K. Nguyen, K. Trieu, and N. Tran (2010), have automated university scheduling using the Tabu Search technique. This approach explores a search space filled with feasible solutions, utilizing memory structures called "Tabus" to avoid revisiting past states and escape local optima. It allows non-improving moves to prevent getting stuck at local maxima. The algorithm effectively navigates complex scheduling landscapes. However, evaluating resources becomes expensive, and problem formulation is notably difficult. These drawbacks limit its widespread practical application despite its effectiveness.
2. N. M. Hussin and A. Azlan (2013), have put into practise the graph coloring heuristic to tackle process scheduling. They modeled events as nodes and conflicts as edges, converting the problem into a graph representation. Domain-specific heuristics are used to order events and assign time slots while ensuring rule compliance. The process includes a building stage for generating solutions and an improvement phase for optimization. While it effectively simplifies the scheduling problem, it does not support soft constraints. Additionally, it takes considerable time to arrive at a solution.
3. W. F. Mahmudy and R. E. Febrita (2017), use fuzzy logic combined with multiple genetic operators for timetable scheduling. Fuzzy logic allows handling uncertainty using approximative rather than exact reasoning, based on degrees of truth between 0 and 1. This helps in reaching realistic and stable solutions more efficiently. It enables smoother transitions and better decision-making in complex scenarios. However, the development of fuzzy logic models and evaluation of membership functions is highly complex. It also demands careful calibration and simulation, making implementation challenging.
4. T. Elsaka (2017), constraint satisfaction modeling for automated timetable generation. This technique emphasizes defining constraints and permissible variable domains rather than optimizing objective functions. It uses backtracking and constraint propagation to reduce domains and find feasible solutions. The approach is clear, flexible, and easy to modify, making it user-friendly for developers. However, it struggles with integrating soft constraints and deeply nested problems. Additionally, identifying a good initial solution can be time-consuming and complex.
5. S. Ab Saad, F. A. Adnan, W. Z. A. Wan Muhamad and Z. R. Yahya (2018), By conducting an investigation, the use of genetic algorithms for solving complex scheduling problems. These algorithms mimic natural selection, where fit individuals are selected iteratively to form optimized schedules. By converting multiple binary constraints into single gene values, problem complexity is significantly reduced. The method is powerful for handling large datasets and diverse variables. However, it is highly parameter-dependent and slow due to its iterative nature. These factors can hinder real-time application in time-sensitive scenarios.

3. METHODOLOGY

The Automatic Timetable Generator follows a modular and layered architecture designed for efficiency, maintainability, and adaptability:

System Architecture

Schedule Sync follows three tier architecture:

- **Frontend**: Built using HTML and CSS, offering a clean, web-based interface for administrators and users to input data and view generated timetables
- **Backend:** Developed in Java (or optionally Python), this layer handles core logic, including data

validation, scheduling rules, and algorithm execution.

- **Database:** MySQL is used to manage essential datasets including faculty details, course assignments, room availability, and student groups.
- **Algorithm Engine:** Implements a Genetic Algorithm for generating optimal, conflict-free timetables based on hard and soft constraints.
- Modules & Libraries: Utilizes custom modules for fitness evaluation, schedule mutation/crossover, and conflict detection.

Workflow

1. **Data Entry and Setup**: Admin inputs data such as faculty names, subjects, classrooms, and student groups through the web interface.
2. **Constraint Definition**: Users define hard constraints (e.g., no overlapping classes, faculty availability) and soft constraints (e.g., preferred timeslots, max lectures per day).
3. **Timetable Generation:** The Genetic Algorithm processes all input data and constraints to generate feasible timetables through evolution-based techniques like selection, crossover, and mutation.
4. **Fitness Evaluation:** Each timetable is scored based on how well it satisfies the defined constraints. Poor solutions are discarded, and better ones evolve over multiple generations.
5. **Output Visualization**: The final timetable is displayed in a user-friendly format on the frontend, allowing for download or printing if needed
6. **Feedback and Adjustment:** Users can review the schedule and, if necessary, adjust constraints or inputs to regenerate the timetable for refinement.

4. Results

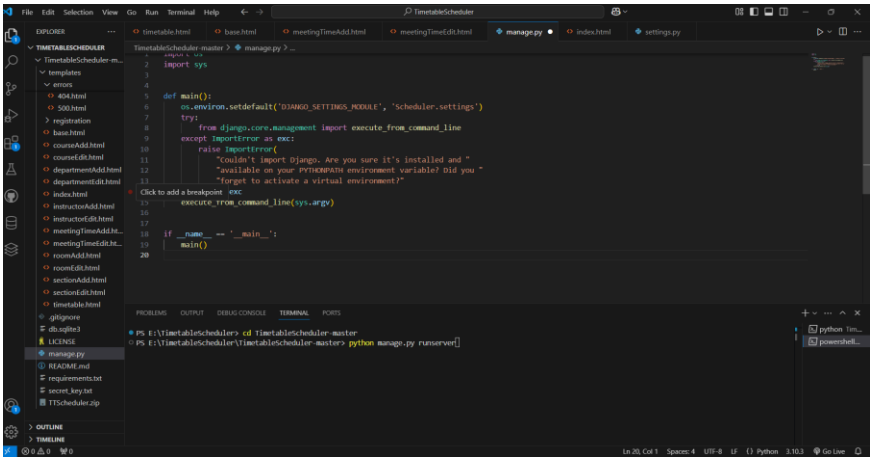

Fig 4.1 Run the web app from the command prompt

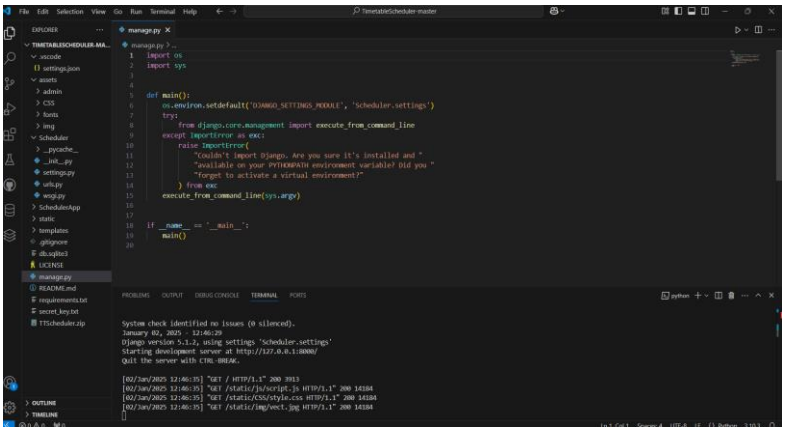

Fig 4.2 URL is generated

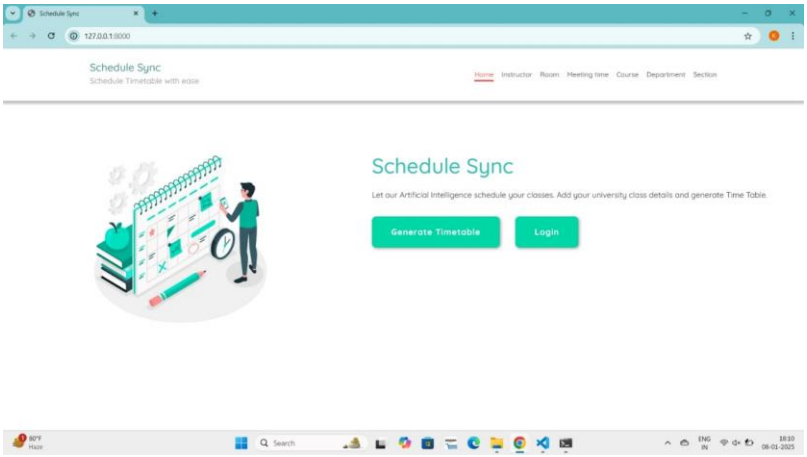

Fig 4.3 web page

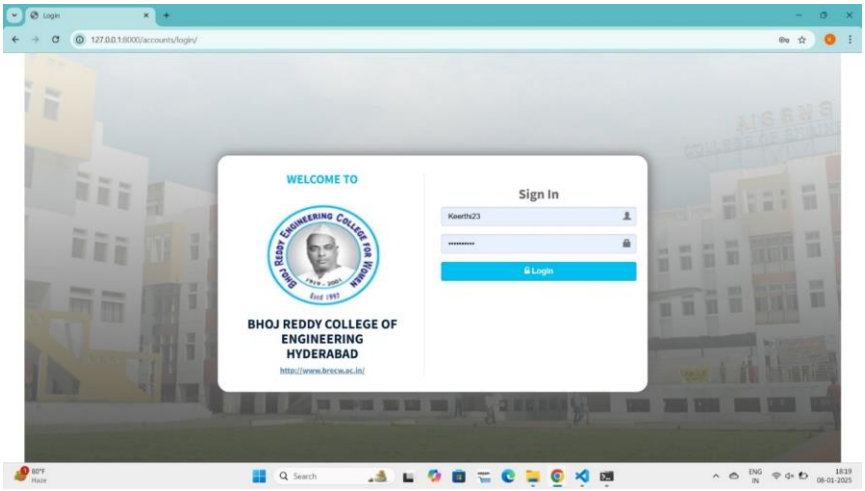

Fig 4.4 Login page

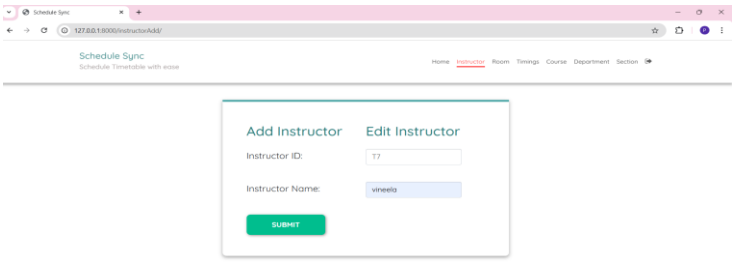

Fig 4.5 Add Instructors

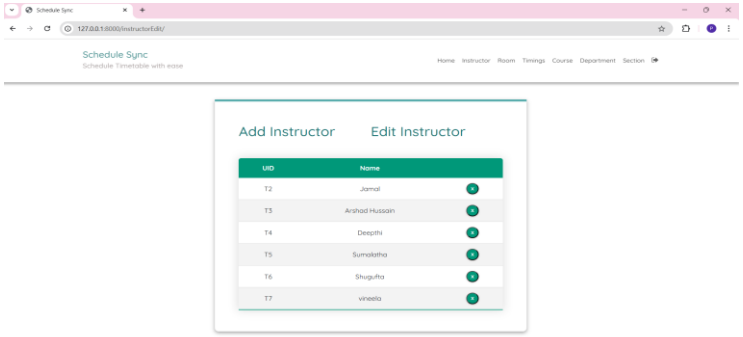

Fig 4,6 Instructor details are uploaded

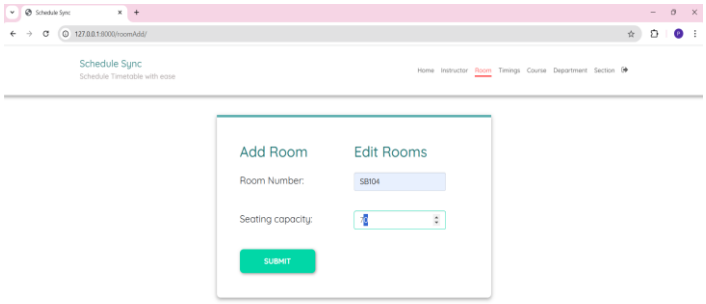

Fig 4.7 Add Rooms

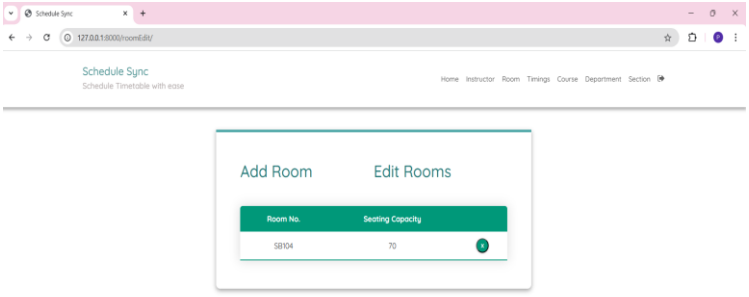

Fig 4.8 Room details are uploaded

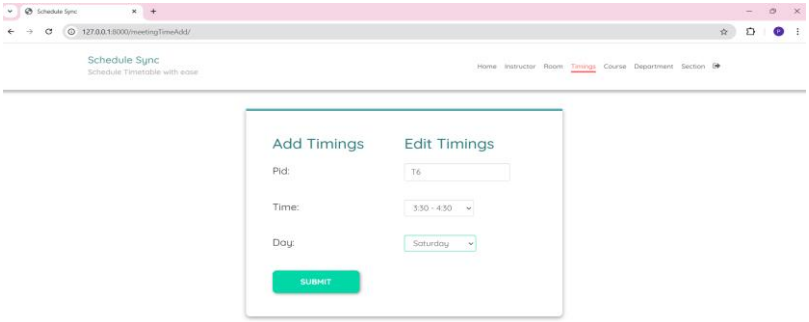

Fig 4.9 Add Timings

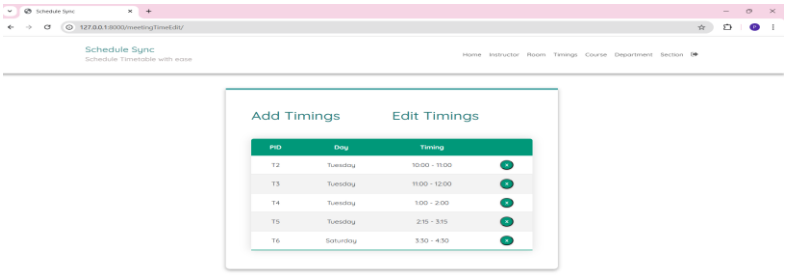

Fig 4.10 Timings details are uploaded

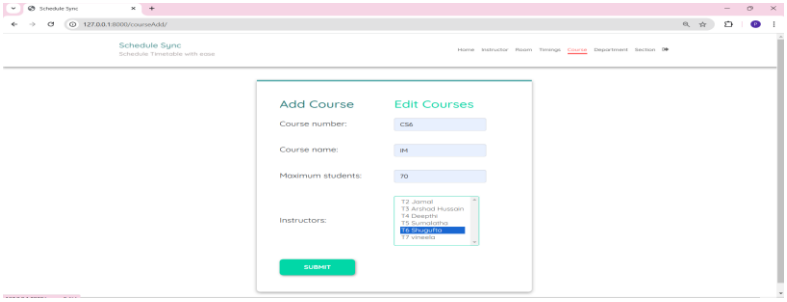

Fig 4.11 Add Courses

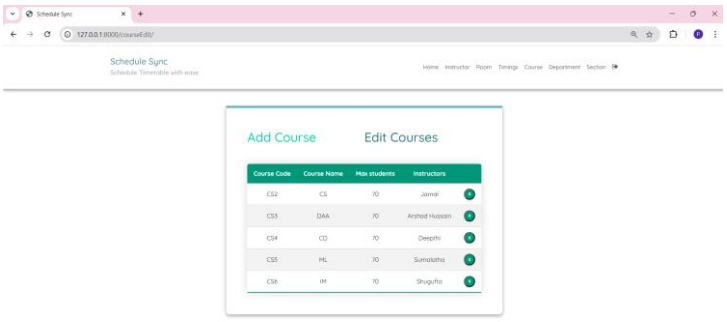

Fig 4.12 Course details are uploaded

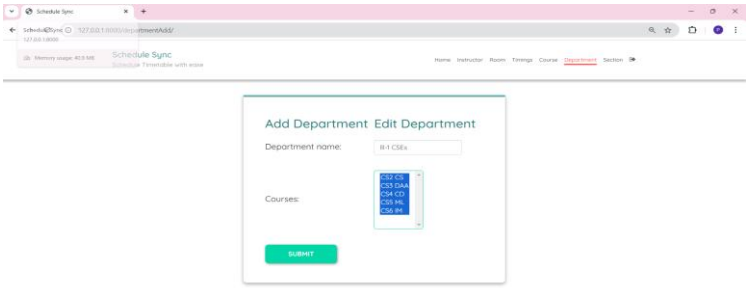

Fig 4.13 Add Departments

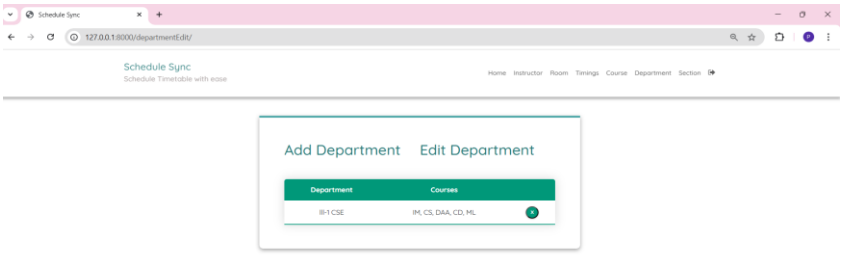

Fig 4.14 Department details are uploaded

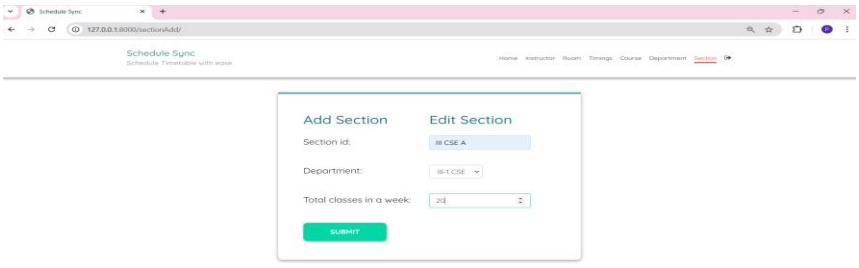

Fig 4.15 Add Sections

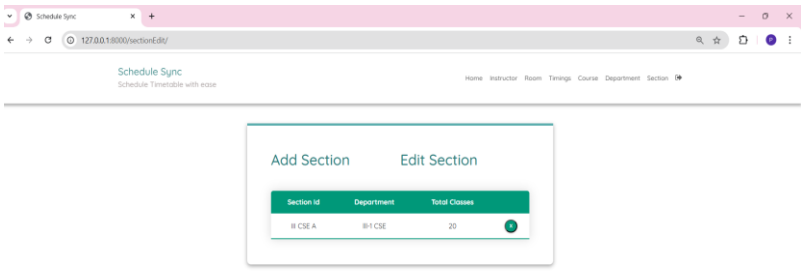

Fig 4.16 Section details are uploaded

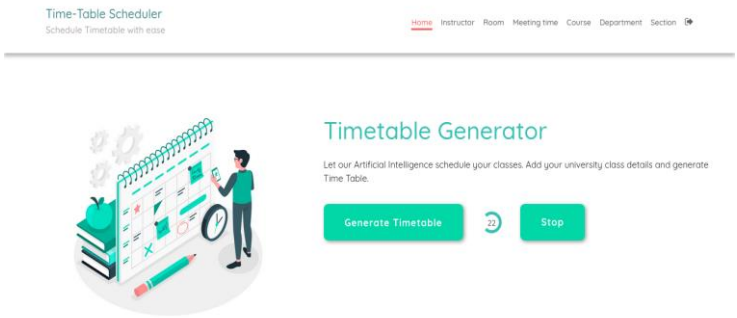

Fig 4.17 Click on generate Timetable

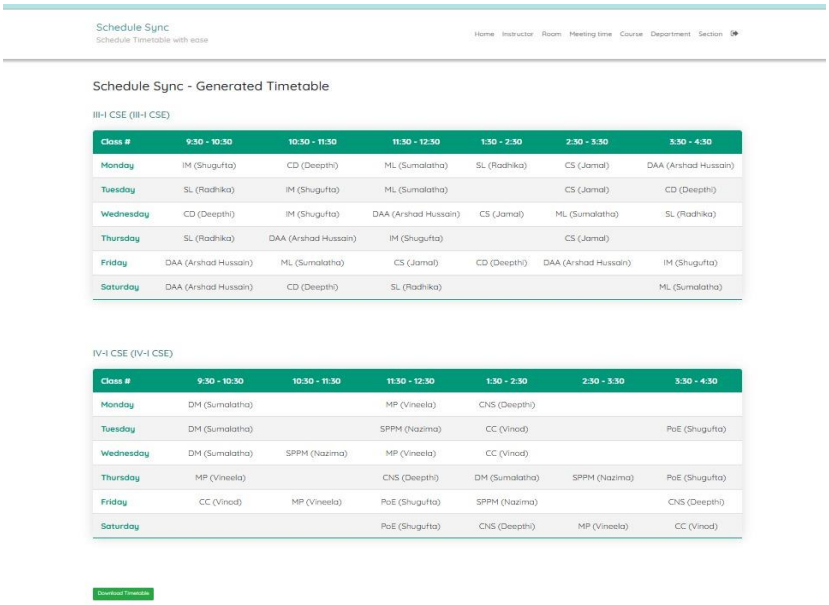

Fig 4.18 Generated Timetables

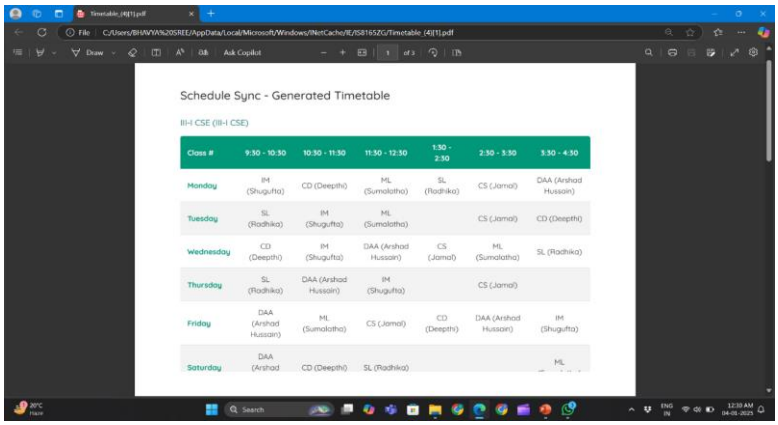

Fig 4.19 Downloaded Pdf

5. CONCLUSION AND FUTURE SCOPE

Conclusion

The Timetable Generator Application successfully automates the complex process of schedule creation, saving significant time and effort for educational institutions. It uses advanced algorithms like backtracking and genetic algorithms to ensure optimized and conflict-free timetables. The system's user-friendly interface simplifies data input and navigation, making it accessible to a broad audience. It accommodates diverse constraints, such as teacher availability and room allocation, ensuring efficient resource utilization. The application's scalability allows it to handle institutions of varying sizes, while its high performance ensures quick and reliable outputs. Overall, the system provides a reliable, efficient, and flexible solution for modern scheduling needs.

Future Scope

The project aims to integrate advanced AI and machine learning techniques to optimize timetables and dynamically resolve scheduling conflicts. It also focuses on developing a mobile application to provide real-time access, instant notifications, and offline availability of timetables. Additionally, the implementation of cloud-based features will enable seamless collaboration, real-time updates, and secure data management.

6. References

[1] K. Nguyen, D. Nguyen, K. Trieu, and N. Tran, "Automating a real-world university timetabling problem with Tabu search algorithm," in 2010 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2010

[2] A. Azlan and N. M. Hussin, "Implementing graph coloring heuristic in construction phase of curriculum-based course timetabling problem," in 2013 IEEE Symposium on Computers & Informatics (ISCI), 2013.

[3] R. E. Febrita and W. F. Mahmudy, "Modified genetic algorithm for high school time-table scheduling with fuzzy time window," in 2017 International Conference on Sustainable Information Engineering and Technology (SIET), 2017.

[4] T. Elsaka, "Autonomous generation of conflict-free examination timetable using constraint satisfaction modelling," in 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), 2017.

[5] F. D. Wihartiko, H. Wijayanti, and F. Virgantari, "Performance comparison of genetic algorithms and

particle swarm optimization for model integer programming bus timetabling problem," IOP Conf. Ser. Mater. Sci. Eng., vol. 332, p. 012020, 2018