WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Schedule Academic Time Table using AI and ML

S.S Sonawane [a], Ram Belitkar [b], Aarya Jambhulkar [c], Raman Yadav [d]

[a] *Professor, Dept. of I.T., AISSMS's Polytechnic, Pune, Maharashtra, India*
[b] *Final Year Student, Department of IT, AISSMS Polytechnic, Pune, Maharashtra, India*
[c] *Final Year Student, Department of IT, AISSMS Polytechnic, Pune, Maharashtra, India*
[d] *Final Year Student, Department of IT, AISSMS Polytechnic, Pune, Maharashtra, India*

A B S T R A C T

This survey paper investigates the fusion of Genetic Algorithms (GAs) and Data Structures and Algorithms (DSAs) to establish a robust Academic Timetable Scheduler empowered by Artificial Intelligence (AI) and Machine Learning (ML) methodologies. The objective of this study is to offer a comprehensive insight into the approaches and tools applied to alleviate the intricacies associated with academic timetable generation. The paper initiates by elucidating the foundational concepts of academic timetable planning and delineating the complexities and constraints inherent in the process. The integration of GAs is explained, emphasizing their ability to enhance the scheduling process through evolutionary principles. A central focus of this survey is the utilization of DSAs to efficiently store and manage academic data. The subsequent section features case studies and real-world applications, illustrating the successful utilization of AI and ML techniques across diverse academic settings. Our main focus in creating this survey is about the time consumed by the Timetable scheduler for returning the output as required, we are going to ensure that the time taken by the timetable scheduler should be less. Additionally, the paper emphasizes the crucial role of JavaScript and other web-based frameworks in delivering user-friendly interfaces and optimal timetable solutions.

Keywords: Academic timetable scheduling, user-friendly interfaces, evolutionary principles, constraints, Real-world applications.

1. INTRODUCTION

In the ever-evolving landscape of educational institutions, effective timetable scheduling remains an enduring hurdle. Acknowledging the pressing demand for inventive solutions, this article introduces a pioneering implementation of Genetic Algorithms to address the intricate Timetable Scheduling Problem. Unlike conventional approaches, our strategy not only seeks to minimize conflicts but also strives to introduce a new era of automated timetable scheduling.

Introducing Genetic Algorithms: Genetic Algorithms, the cornerstone of our revolutionary approach, leverage the principles of natural selection to optimize intricate problems. Within this context, we delve into the essence of Genetic Algorithms, illuminating their inner workings and demonstrating their flexibility in resolving complex scheduling challenges.

Challenges in Timetable Management: Educational institutions wrestle with the complexities of timetable management, navigating a myriad of conflicts and constraints. Our objective is to dissect the current issues affecting timetable management, clearing the path for an innovative paradigm shift.

Project Objectives: Unveiling Genetic Algorithms: Uncover the mysteries surrounding Genetic Algorithms, offering a comprehensive insight into their mechanisms and applicability.

Challenges in Timetable Management: Investigate the existing hurdles in timetable management, underscoring the necessity for a transformative approach to scheduling.

Prototype for Automated Timetable Scheduling: Display the capabilities of Genetic Algorithms through the development of a prototype for an automated timetable scheduling system. This signifies a departure from traditional methods, presenting a fresh solution to longstanding problems.

2. LITERATURE SURVEY

[1] P. Uma: Assistant Professor, Computer Science and Engineering, Nanda Engineering College, Erode. P. S. Sharvesh: Student, Computer Science and Engineering, Nanda Engineering College, Erode. M. Pradeep: Student, Computer Science and Engineering, Nanda Engineering College, Erode.

P. Sathishkumar: Student, Computer Science and Engineering, Nanda Engineering College, Erode. R. Senthilnathan: Student, Computer Science and Engineering, Nanda Engineering College, Erode. Automation significantly reduces the time and effort required to create schedules, improving the overall

efficiency of the scheduling process. The use of multiple algorithms allows for optimization and the generation of high-quality timetables that adhere to various constraints and objectives. The project aims to efficiently utilize all available resources, which is essential for educational institutions. Implementing multiple algorithms for timetable generation can be complex and resource-intensive, requiring careful design, integration, and evaluation. The success of the system may depend on how well it aligns with the needs and preferences of the educational institution's faculty and administration. Develop an intuitive and user-friendly interface that allows users to input scheduling constraints and preferences easily. Implement a system that allows real-time changes and rescheduling based on evolving constraints, faculty availability, and student preferences. Prioritize data security and privacy, especially when handling sensitive academic and student information. Integrate a feedback mechanism for users to report issues, suggest improvements, and provide input on scheduling preferences.[1]

[2] Mr. M. K. Mohamed Faizal: Assistant Professor, M.I.E.T. Engineering College Trichy, India. Vanmathi Karunanithi: Computer Science and Engineering, M.I.E.T. Engineering College Trichy India. Thamarai Selvi Balu: Computer Science and Engineering, M.I.E.T. Engineering College Trichy India. Nithya Senthilkumar: Computer Science and Engineering, M.I.E.T. Engineering College Trichy India. The project mentions using a data category algorithm to categorize department-wise time scheduling, which can help optimize class timings based on specific requirements and constraints. By preventing overlapping schedules and making the best use of faculty subject demands, the system can help optimize resource utilization within the institution. An automated timetable generation system can significantly reduce the time and effort required to create schedules, improving the efficiency of the scheduling process. The project scalability to accommodate scheduling challenges in larger institutions may not be clear. The effectiveness of the solution may vary based on the size and complexity of the college. Implementing an effective automated timetable generation system can be complex, particularly when dealing with a wide range of scheduling constraints and preferences. Develop an intuitive user interface that allows for easy input of scheduling constraints and preferences. This can enhance the accessibility and user-friendliness of the system. Integrate feedback mechanisms for users to report issues, suggest improvements, and provide input on scheduling preferences. Prioritize data security and privacy, especially when dealing with sensitive academic and student information.[2]

[3] Prof. Jyothi Patil1, Shambhavi V2, Sneha N T3, Sweta Jadhav4, Tahura Sadaf5 Department of Electronics & Communication Engineering, P D A College of Engineering, Kalaburagi, Karnataka, India. The goal of this project is to create a timetable generator for colleges. The creation of schedules is a very common issue that affects all educational institutions. The conflict between staff members' preferences is precisely where the issue arises. Every semester, colleges are required to create timetables, which used to be an extremely time-consuming task. Once the timetables are set for a given semester, the student is allowed to access them. Once the timetables are established for a particular semester, employees are also permitted to check the class allotment schedule. The Time Table Assignment for Any Department project's goal was to create an application that would allow staff and student allotment subject to classes. The following information was added by the administrator for Adding the student, the staff, and the subject, enter the timetable, and update the timetable. The majority of colleges offer a variety of programmes, each of which has several disciplines. There are now a limited number of faculties, each of which teaches many disciplines. Therefore, the timetable now has to include the instructors at the appropriate times. the timetable schedule, which makes the most use of all faculty subject demands, and slots so that their timings do not cross. For this, a genetic algorithm is employed. We suggest using a timetable object in our method for creating timetables. This object consists of classroom objects, their respective schedules, and a fitness rating for the schedule. Additionally, to further describe the imperatives, we used a composite configuration design that is easily expandable to include or uproot as many duties. Every obligation class now checks the condition found in our investigation between two timetable objects. In the unlikely event that the requirement is met, the score is raised by one if a crash is available.[3]

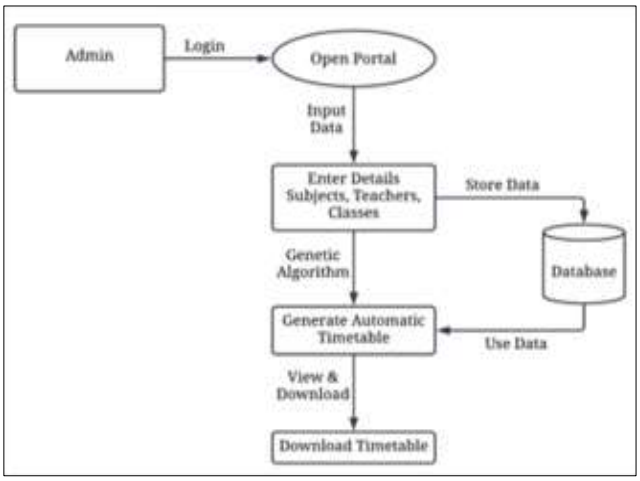
[4] Authored by Tiny Wijerathna Ekanayake, Pavani Subasinghe, Shawn Ragel, Anjalie Gamage, and Suchini Attanayaka from the Faculty of Computing at Sri Lanka Institute of Information Technology, Malabe, Sri Lanka. This project explicitly delves into the domain of two distinct timetables: examination and semester timetables, acknowledging the inherent differences in constraints and requirements associated with each type. Timetable scheduling presents a concrete real-world challenge in educational institutions, with potential solutions directly impacting efficiency and resource allocation. The intricate process of implementing and comparing multiple scheduling algorithms necessitates a thoughtful blend of careful design and meticulous evaluation. The effectiveness of these algorithms depends on the specific rules, constraints, and requirements unique to different educational institutions, making a universally applicable solution challenging to attain.

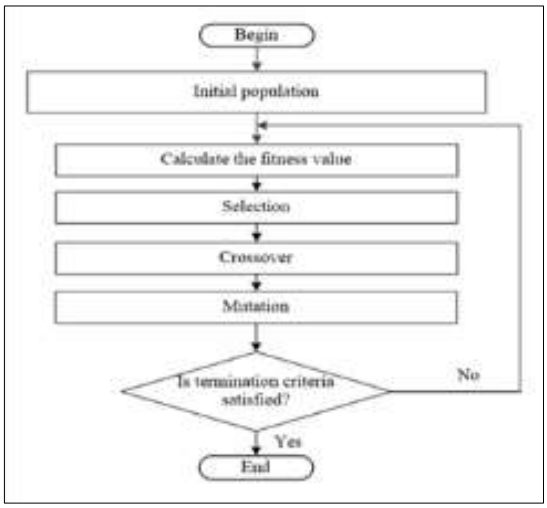
Recognizing the absence of a one-size-fits-all solution, the project emphasizes the importance of developing an intuitive user interface. This interface serves as an entry point for educational institutions to easily input their specific constraints and preferences. Furthermore, the algorithms employed should demonstrate scalability, capable of addressing scheduling challenges for educational institutions varying in size and complexity. Essentially, the Intelligent Timetable Scheduler represents a comprehensive exploration into diverse algorithms, aiming to pave the way for customized solutions in the domain of educational timetable management.[4]

[5] Ernesto Cortés Pérez: Ingeniería en Computacion, Universidad del Istmo, UNISTMO, Sto. Domingo Tehuantepec, Oaxaca, México. Osiris Montero Rios: Ingeniería en Computacion, Universidad del Istmo, UNISTMO, Sto. Domingo Tehuantepec, Oaxaca, México. Daniel Pacheco Bautista: Ingeniería en Computacion, Universidad del Istmo, UNISTMO, Sto. Domingo Tehuantepec, Oaxaca, México. Sergio Sanchez Sanchez: Ingeniería en Computacion, Universidad del Istmo, UNISTMO, Sto. Domingo Tehuantepec, Oaxaca, México. Francisco Aguilar Acevedo: Ingeniería en Computacion, Universidad del Istmo, UNISTMO, Sto. Domingo Tehuantepec, Oaxaca, México. Using a Genetic Algorithm can significantly reduce the time and effort required to create university timetables, particularly when dealing with many professors, subjects, and constraints. Genetic Algorithms have the potential to find high-quality schedules that meet various constraints and objectives, contributing to better resource allocation and utilization. The project addresses a real-world problem that many educational institutions face, making it directly relevant and impactful. The solution may not be directly applicable to other educational institutions as scheduling constraints and requirements can vary significantly. Implementing a Genetic Algorithm for timetable scheduling

requires computational resources and expertise in algorithm design and optimization. Implementing a Genetic Algorithm for timetable scheduling requires computational resources and expertise in algorithm design and optimization. Integrate a feedback mechanism for users to report issues and suggest improvements, helping in ongoing refinement of the scheduling algorithms. Prioritize data security and privacy, especially when dealing with sensitive academic and student information.[5]

3. PROPOSED METHODOLOGY





Explanatory Description:

Module 01: Input Data

Module Purpose:

The primary objective of Module 01 is to gather and compile essential input data required for creating a comprehensive training dataset. This **dataset will** serve as the backbone for training the algorithms utilized in the Academic Timetable.

Data Collection:

Initiate the process by gathering pertinent data from various sources including teachers, courses, and available rooms. These data should accurately represent the entities involved in the academic scheduling process.

Data Quality Assurance:

Ensuring data quality is paramount. Verify that the collected information is accurate, up-to-date, and comprehensive, thereby reflecting the diverse range of academic activities and resources.

Data Annotation:

Annotate each piece of data with relevant metadata such as teacher names, course codes, and room capacities. This step is crucial for facilitating algorithmic processing during timetable generation.

Data Augmentation (Optional):

Consider employing data augmentation techniques if necessary to enrich the dataset. These techniques may involve generating additional data points by manipulating existing information or introducing synthetic data to simulate real-world scenarios.

Privacy and Security:

Uphold stringent measures to safeguard data privacy and security. Obtain consent from relevant stakeholders for the use of their information and implement robust security protocols to prevent unauthorized access.

Balanced Representation:

Strive to achieve a balanced representation of different entities within the dataset, including teachers, courses, and available rooms. Imbalances may skew the scheduling outcomes and lead to inefficiencies.

Dataset Management:

Establish a well-organized dataset management system to efficiently manage, store, and retrieve the input data. Document the dataset structure, update protocols, and any modifications made over time.

Regular Updates:

Acknowledge the dynamic nature of academic institutions and anticipate the need for periodic updates to the dataset. Incorporate mechanisms to accommodate changes in faculty, courses, and facilities.

Compatibility with Timetable Generation Algorithms:

Ensure compatibility between the input data and the algorithms chosen for timetable generation. This may entail data formatting, pre-processing, or customization to align with algorithmic requirements.

Backup and Redundancy:

Implement robust backup and redundancy mechanisms to mitigate the risk of data loss and ensure continuous availability of the dataset throughout the project lifecycle.

Ethical Considerations:

Adhere to ethical guidelines and regulatory requirements governing data handling, particularly concerning the confidentiality and consent of individuals represented in the dataset. Prioritize the protection of privacy and rights throughout the data lifecycle. Module

02: Real-time Time Scheduling and Data Utilization

1. **Real-Time Scheduling Engine:**
    - Implement a real-time scheduling engine capable of dynamically adjusting timetables based on changing circumstances and requirements.
2. **Dynamic Resource Allocation:**
    - Develop algorithms and rules for allocating classrooms, faculty, and other resources in real time to minimize conflicts and optimize resource utilization.
3. **Data Integration:**
    - Ensure seamless integration of academic data, including course schedules, faculty availability, and student preferences, into the real-time scheduling engine.
4. **Live Data Feeds:**
    - Establish mechanisms for live data feeds, allowing the system to receive and process real-time updates on class cancellations, room availability, and faculty substitutions.
5. **Adaptive Decision-Making:**
    - Implement intelligent decision-making processes that consider multiple factors, such as room capacities, faculty expertise, and student enrolment, to make scheduling decisions in real-time.
6. **Conflict Resolution Logic:**
    - Develop algorithms and logic for conflict resolution, enabling the system to address scheduling conflicts as they arise.
7. **User Notifications:**
    - Configured the system to send real-time notifications to educators, students, and administrators regarding schedule changes, cancellations, and updates.
8. **Data Analytics and Insights:**
    - Implement data analytics tools to generate insights from the academic data, helping institutions make data-driven decisions for timetable adjustments.
9. **Feedback Loop:**
    - Create a feedback mechanism for users to report scheduling issues and preferences, allowing the system to continuously adapt and improve.

Requirement Analysis:

Functional Requirements

1. Functional Requirements

2. Non – Functional Requirements

Functional Requirements:

1.User Interface Implementation:

Implement a user interface for the system, allowing users to interact with the application seamlessly.

2.User-Friendly Front-End Design:

Develop a front-end design using Cascading Style Sheets (CSS) to ensure a visually appealing and intuitive user experience.

3.Strong Authentication:

Implement robust authentication mechanisms to ensure secure access to the system, requiring users to authenticate themselves before performing any operations.

4.JavaScript Validations and Alerts:

Incorporate JavaScript validations and alerts at appropriate places within the user interface to provide real-time feedback and guidance to users during data entry and interactions**.**

Non-Functional Requirements:

Secure Access to Confidential Data:

Ensure secure access to confidential user data (e.g., user details) by implementing SSL (Secure Sockets Layer) encryption for data transmission over the network.

Better Component Design for Performance:

Design system components efficiently to optimize performance, particularly during peak usage times. This may involve architectural considerations such as caching, load balancing, and efficient database queries.

Flexible Service-Based Architecture:

Design the system with a flexible service-based architecture to facilitate easy integration of future extensions or enhancements. This architecture should allow for seamless scalability and modularity, enabling the addition of new features or functionalities without significant disruption to existing systems.

Implementation Details:

Back-End: Python is used for the back-end development, leveraging its robustness, versatility, and extensive library support.

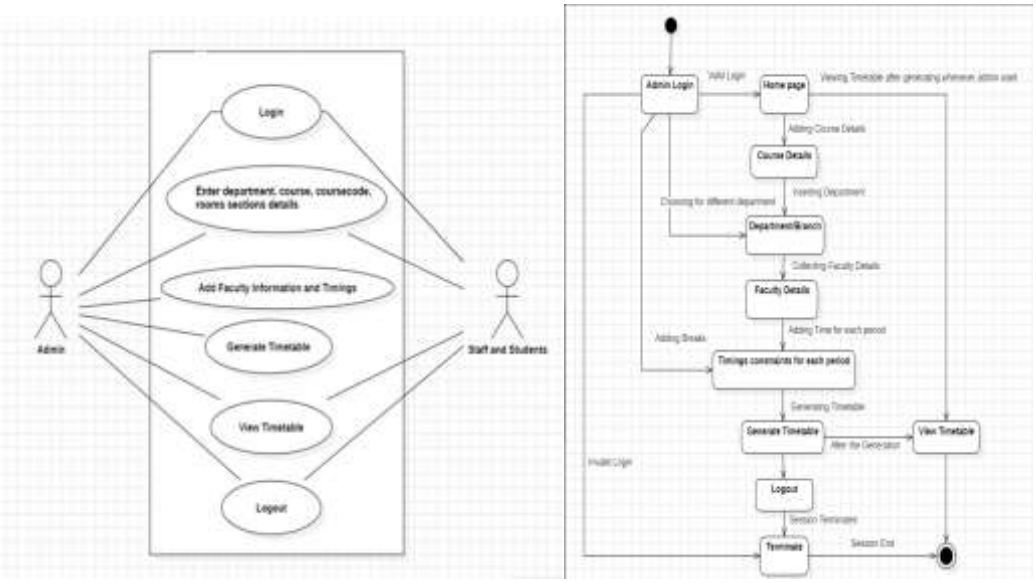
Front-End: HTML, CSS, and JavaScript are utilized for front-end development, ensuring a dynamic and interactive user interface.

Framework: Django is employed as the web framework for connecting the front-end and back-end components, streamlining development and providing built-in security features.

Database: SQLite3 is chosen as the database management system for its simplicity and ease of integration with Django, suitable for smaller-scale applications.

1.1 Design Specification & system functioning

Uml Diagrams



Explanation of use case diagram

Register User:

Actors: New User

Description: New users (e.g., administrators, teachers, students) can register for an account by providing necessary information such as name, email, and password.

Login:

Actors: Registered User

Description: Registered users can log in to the system using their credentials (e.g., email and password) to access the system's functionalities.

Manage Teachers:

Actors: Administrator

Description: Administrators can add, modify, or delete teacher profiles, including information such as name, contact details, expertise, and teaching preferences.

Manage Courses:

Actors: Administrator

Description: Administrators can manage courses offered by the college, including adding new courses, modifying existing ones, and specifying course details such as name, code, credits, and prerequisites.

Manage Departments:

Actors: Administrator

Description: Administrators can manage academic departments within the college, including adding new departments, modifying existing ones, and assigning department heads.

Manage Professors:

Actors: Administrator

Description: Administrators can manage professor profiles, including adding new professors, modifying existing ones, and assigning professors to specific departments and courses.

Generate Timetable:

Actors: Administrator

Description: The administrator triggers the system to generate a new timetable using the genetic algorithm. The system utilizes the genetic algorithm subsystem to create an optimized timetable based on input constraints and preferences.

View Timetable:

Actors: Administrator, Teachers, Students

Description: Users can view the generated timetable. Administrators, teachers, and students may have different access levels and views of the timetable based on their roles and permissions.

Modify Timetable:

Actors: Administrator

Description: The administrator has the capability to make manual adjustments or modifications to the generated timetable if necessary. This may include resolving conflicts, accommodating special requests, or making last-minute changes.

Evaluate Timetable:

Actors: Administrator

Description: The administrator evaluates the quality and effectiveness of the generated timetable. This may involve assessing factors such as room utilization, teacher preferences, student satisfaction, and overall timetable efficiency.

4. Proposed Algorithm

The whole method of scheduling based on a genetic algorithm is explained in detail in this section.

A scheduling procedure is divided into several important modules as follows,

1. Data encoding and decoding
2. Initial population
3. Evaluation of population
4. Crossover Evolution
5. Mutation
6. New Population

Data Encoding and Decoding:

Data encoding is the initial step in the Genetic Algorithm, transforming a solution into a chromosome represented as a simple value, such as a string. This process enhances algorithm speed, and a common approach is to convert data into a binary string. Genes, constituting parts of chromosomes, can also be converted to binary strings. This type of data conversion facilitates algorithm treatment, with the chromosome string composed of side-by-side gene strings.

Initial Population:

The first step in the Genetic Algorithm involves creating multiple random individuals based on hard constraints. The size of the initial population is determined by user needs. A small population may diminish over generations due to evolution, while a large population yields better results at the cost of increased resources and slower processing. The population is often represented as a set

Evaluation of Population:

The fitness of a solution is evaluated using soft constraints, determining how good the solution is within a valid range. Population evaluation is a crucial aspect of the Genetic Algorithm, employing a fitness function to rank solutions. Fitness is usually expressed on a scale of 0 to 1, where 1 represents the best solution, and 0 represents the worst. This allows for a comparative assessment of solutions within the population**.**

Crossover Evolution:

Crossover evolution is a method for generating a new population based on the existing one. The simple crossover evolution involves taking two chromosomes and creating a specified number of new chromosomes (X). This process entails splitting the two chromosomes into parts and generating new chromosomes by combining different parts. The goal is to introduce diversity and potentially enhance solution quality.

Mutation:

Mutation is employed to introduce randomness and variability into the algorithm. It involves randomly changing the values of a gene, leading to the creation of new, unexpected solutions. These mutated solutions provide a different perspective for the fitness function. Importantly, mutation only affects the chromosome undergoing the mutation, leaving other solutions unchanged.

New Population:

The combination of crossover and mutation operations results in the creation of a new population comprising original solutions. This new population inherits traits from the previous generation while introducing diversity through genetic recombination and random mutations.potential integration of IoT devices for additional functionality are avenues for improvement, with ongoing compliance with data privacy regulations being essential for ethical and legal standing. Pursuing these developments can make the Real-Time Attendance System a versatile and indispensable tool in educational technology, contributing to efficiency and engagement.

5.Results & output

Fig:- Login Page

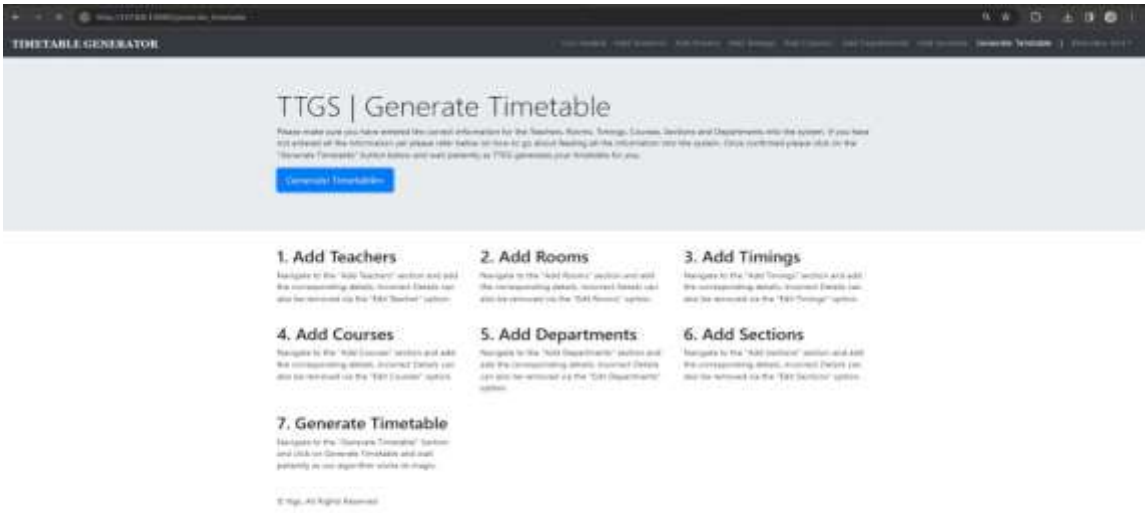


Fig:- Home/Navigation Page

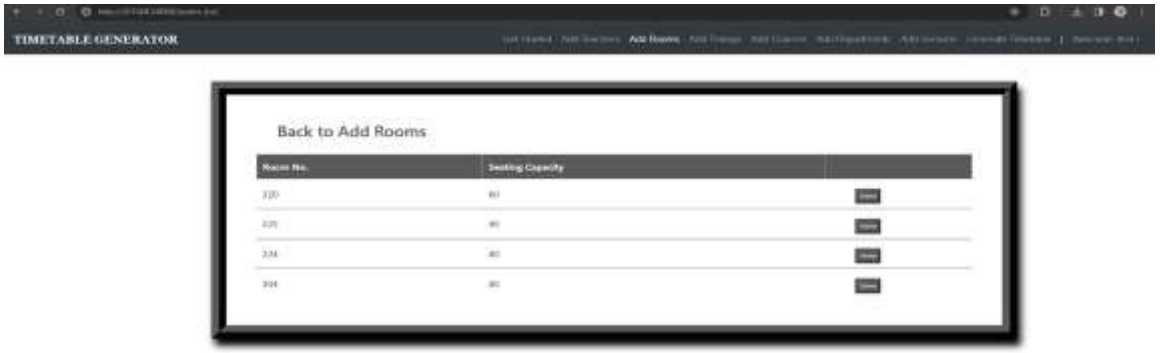


Fig:- Add Rooms

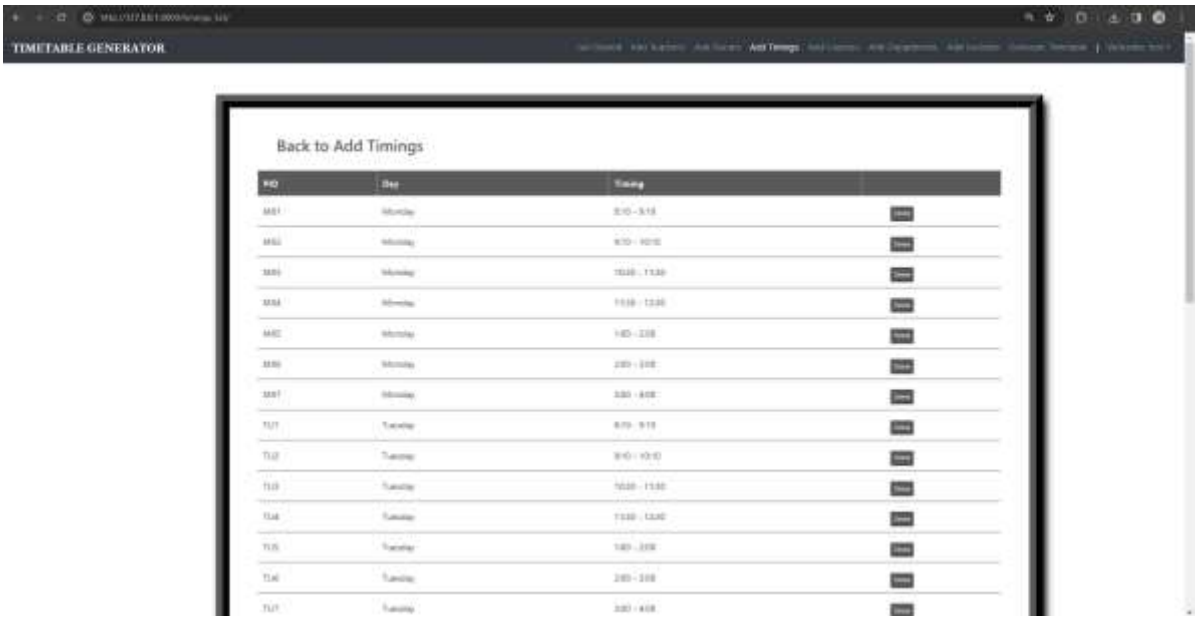


Fig:- Add Timings

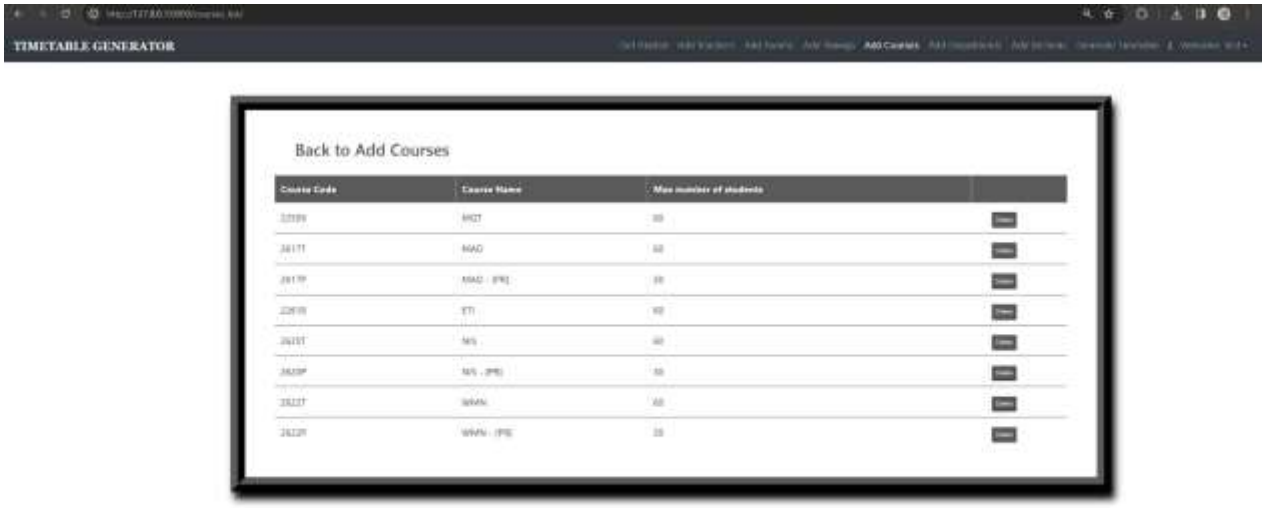


Fig:- Add Courses

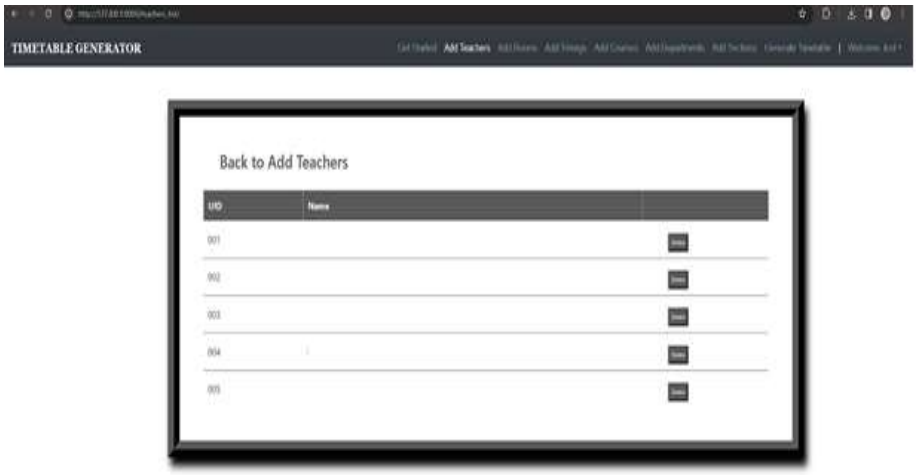


Fig:- Add Teachers

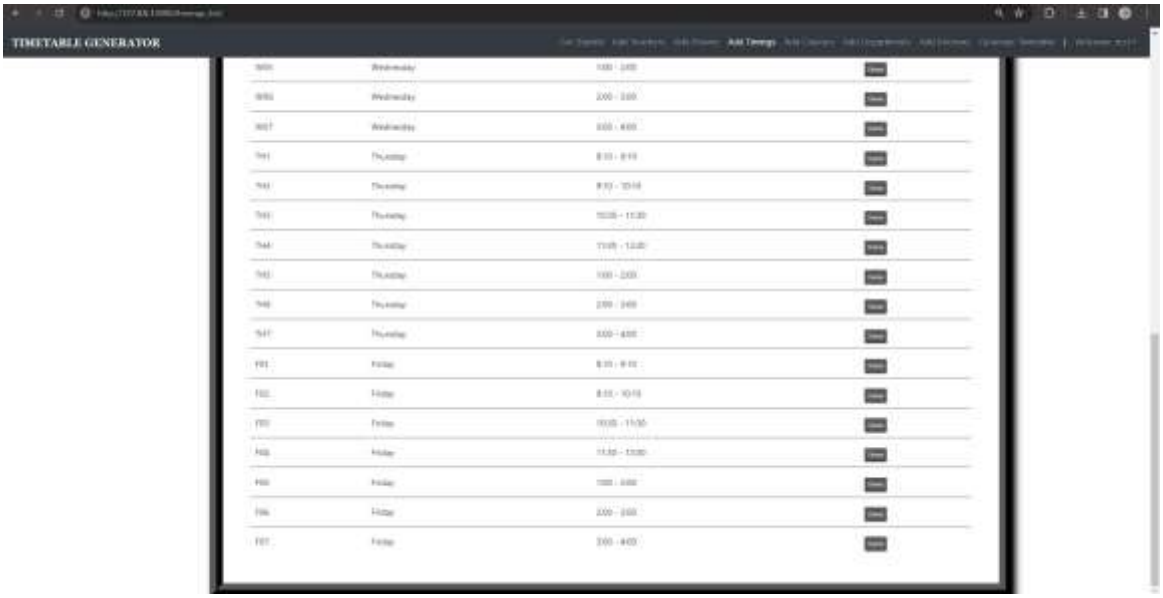


Fig:- Final output

6. Conclusion

In conclusion, the implementation of the "Schedule Academic Timetable using Genetic Algorithm" project represents a significant advancement in the realm of academic timetable management. With a strategic approach to planning and execution throughout each phase of the project, we have witnessed a remarkable evolution in how academic institutions handle scheduling processes.

The notable achievements of this initiative underscore the effectiveness of employing genetic algorithms in optimizing academic timetables. By leveraging this sophisticated approach, we have successfully addressed the complexities inherent in scheduling courses, allocating resources, and accommodating various constraints within academic environments. Through the systematic application of genetic algorithms, we have achieved a level of efficiency and optimization that was previously unattainable through manual methods.

Furthermore, this project has demonstrated the potential for technology, specifically genetic algorithms, to profoundly impact academic administration. It not only streamlines the scheduling process but also enhances the overall effectiveness of academic institutions by ensuring optimal resource allocation and minimizing scheduling conflicts. As we reflect on the outcomes of this implementation, it becomes evident that the adoption of genetic algorithms in academic timetable scheduling represents a transformative step forward. It marks a departure from traditional approaches towards a more data-driven and algorithmically optimized methodology. In essence, the "Schedule Academic Timetable using Genetic Algorithm" project serves as a testament to the potential of technology to revolutionize academic administration. Moving forward, it is imperative to continue exploring innovative solutions and refining existing methodologies to further enhance the efficiency and effectiveness of academic timetable management.

References

[1] “Automatic Timetable Generation”

P. Uma: Assistant Professor, Computer Science and Engineering, Nanda Engineering College, Erode. P.

S. Sharvesh: Student, Computer Science and Engineering, Nanda Engineering College, Erode. M.

Pradeep: Student, Computer Science and Engineering, Nanda Engineering College, Erode. P.

Sathishkumar: Student, Computer Science and Engineering, Nanda Engineering College, Erode.

R. Senthilnathan: Student, Computer Science and Engineering, Nanda Engineering College, Erode.

Publisher Name: IJRASET (2023-04-05).

[2] “College Timetable using Time Scheduling Algorithm”

Mr. M. K. Mohamed Faizal: Assistant Professor, M.I.E.T. Engineering College Trichy, India. Vanuatu

Karunanithi: Computer Science and Engineering, M.I.E.T. Engineering College Trichy, India. Thamarai

Selvi Balu: Computer Science and Engineering, M.I.E.T. Engineering College Trichy, India. Nithya

Senthilkumar: Computer Science and Engineering, M.I.E.T. Engineering College Trichy India.

Publisher Name: IJERT (30-07-2022).

[3] "Automatic Timetable Generator"

Prof. Jyothi Patil1, Shambhavi V2, Sneha N T3, Sweta Jadhav4, Tahura Sadaf Department of Electronics & Communication Engineering, P D A College of Engineering, Kalaburagi, Karnataka, India.

Publisher Name: IJRASET (2023-06-07).

[4] "Intelligent Timetable Scheduler: A Comparison of Genetic, Graph Coloring, Heuristic and Iterated Local Search Algorithms"

Tiny Wijerathna Ekanayake: Faculty of Computing, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka. Pavani Subasinghe: Faculty of Computing, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka. Shawn Ragel: Faculty of Computing, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka. Anjalie Gamage: Faculty of Computing, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka. Suchini Attanayaka: Faculty of Computing, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka.

Publisher: IEEE (2019 International Conference on Advancements in Computing (ICAC)).

[5] "A Genetic Algorithm Solution for Scheduling Problem"

Ernesto Cortés Pérez: Ingeniería en Computación, Universidad del Istmo,

UNISTMO, Sto. Domingo Tehuantepec, Oaxaca, México. Osiris Montero Rios: Ingeniería en Computación, Universidad del Istmo, UNISTMO, Sto. Domingo

Tehuantepec, Oaxaca, México. Daniel Pacheco Bautista: Ingeniería en Computación, Universidad del Istmo, UNISTMO, Sto. Domingo Tehuantepec,

Oaxaca, México. Sergio Sanchez Sanchez: Ingeniería en Computación,

Universidad del Istmo, UNISTMO, Sto. Domingo Tehuantepec, Oaxaca, México. Francisco Aguilar Acevedo: Ingeniería en Computación, Universidad del Istmo, UNISTMO, Sto. Domingo Tehuantepec, Oaxaca, México.

Publisher: IEEE (2021 XVII International Engineering Congress (CONIIN)).