International Journal of Research Publication and Reviews

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

Summer-Term Timetable Generation

S.Pavani[1], Thummalapalle Vamshika[2], Meda Sai Srihitha[3]

[1]UG Student SOCSE, Presidency University, Banglore-560 064, India, pavani.2003workmail@gmail.com.
[2]UG Student SOCSE, Presidency University, Banglore-560 064, India, tvamshika99@gmail.com
[3]UG Student SOCSE, Presidency University, Banglore-560 064, India, medasrihitha97@gmail.com

ABSTRACT:

Creating academic schedules is a difficult optimization task, especially during the summer when special limitations like shorter course lengths, erratic teacher availability, and scarce physical resources are introduced. Conventional manual scheduling techniques are laborious, ineffective, and prone to disagreement, particularly during the shortened summer session schedule. In order to overcome these obstacles, this work explores the use of sophisticated algorithmic techniques such as machine learning, metaheuristics, and constraint satisfaction. It offers a thorough examination of the scheduling needs for the summer, assesses the effectiveness of several optimization strategies, and suggests a hybrid model that combines adaptive metaheuristics and rule-based restrictions. With possible ramifications for larger scheduling settings, the findings offer useful methods for enriching the academic experience, optimizing resource allocation, and lowering administrative labor.

INTRODUCTION

The majority of realistic circumstances classify timetable creation as NP-hard because of its well-known computational difficulty. Even though they are still widely used in many academic settings, traditional manual scheduling methods have a number of drawbacks. They can be very time-consuming (requiring weeks of administrative work), inefficient in allocating resources, and frequently result in scheduling conflicts that can negatively impact both faculty and students. During the summer term, when there is less time for scheduling adjustments due to the abbreviated timeframe, these problems become even more urgent.

There is hope that recent advancements in algorithmic techniques may help address these issues more successfully. Constraint satisfaction approaches, for instance, provide a methodical way to incorporate physical and institutional constraints, while sophisticated methods such as ant colony optimization, simulated annealing, and genetic algorithms have shown promise in exploring the wide variety of possible schedules. More recently, real-time resource distribution optimization and course demand trend forecasting have been achieved with the use of machine learning techniques. However, current research tends to focus on traditional semester systems and has not fully explored the utility of these strategies specifically for scheduling summer terms.

Three major contributions to the field of academic scheduling are made by this study. First, with the help of case studies from different institutions, it provides a thorough analysis of the particular requirements and limitations related to summer term scheduling. Second, it evaluates how well various algorithmic techniques work in summer session settings, gauging their effectiveness in terms of solution quality, computing speed, and usefulness. Third, the study presents a hybrid optimization model that successfully addresses the dynamic scheduling issues of summer terms by combining rule-based restrictions with adaptive metaheuristics. The study's findings have applications beyond computer science theory, offering university managers who want to improve operational efficiency throughout the summer months a useful edge. Institutions can enhance resource allocation, reduce administrative burdens, and ultimately improve the educational experience for professors and students during these concentrated academic periods by developing more efficient scheduling systems. Additionally, as many strategies created for handling busy schedules and scarce resources are applicable in broader contexts, the knowledge gathered from summer term scheduling could result in improvements in the creation of regular semester timetables.

LITERATURE REVIEW

[1] Khalil, M., and Hassan, M. (2023) Title: A Machine Learning-Based Intelligent Course of the Journal of Educational Computing Research. Methodology Suggestion: Hassan and Khalil suggested an automated and adaptive method for managing academic course scheduling that is based on machine learning. In order to train models that can forecast ideal timetables, the system uses supervised learning techniques, which necessitate a carefully selected dataset. The system constantly modifies schedules in accordance with institutional policies and limits, taking into account variables such as student demands, teacher preferences, and course availability. Benefits:able to adjust to changing academic settings. Restrictions: reliance on large, high-quality datasets for efficient training. Limited relevance for organizations without adequate data access.

[2] Wang, X., & Xu, H. (2021) Paper: Expert Systems with Applications.Title: A New Memetic Algorithm for Resolving University Scheduling Issues Proposed Methodology: To improve timetable optimization, Wang and Xu presented a memetic algorithm that combines local search methods with evolutionary algorithms. Using a genetic algorithm, this hybrid strategy first looks globally for workable solutions before making localized adjustments to enhance schedule quality. Even in the face of intricate scheduling requirements, the memetic structure permits flexibility and optimization.Benefits: Excellent quality and efficiency in resolving complex scheduling problems. Restrictions: high demand for computational resources, which limits its usefulness in environments with limited resources.

[3] Pillay, N., & Qu, R. (2022) Title: An Evolutionary Algorithm for the Multi-Criteria University Timetabling Problem.Proposed Methodology: Faculty preferences, student needs, and institutional constraints are all taken into account in this study's multi-criteria evolutionary algorithm. The method explores a large solution space and converges on ideal timetables that fulfill a variety of objectives by employing evolutionary processes including mutation and crossover.

Benefits: adaptability in striking a balance between various institutional elements. Restrictions: Implementation necessitates intricate parameter adjustment, which makes deployment and maintenance more difficult.

[4] Le, M. T., and Nguyen, T. T. (2021).Title: An Automated Course Scheduling Method Based on Deep Reinforcement Learning Published by IEEE Access, Volume 9, Pages 115765–11577. Suggested Approach: To automate course scheduling, Nguyen and Le used deep reinforcement learning (DRL). Their approach uses predetermined limitations and previous data to teach an agent the best scheduling policies. By interacting with the environment and getting feedback, the agent iteratively optimizes schedules. Benefits: adapts dynamically to past patterns for increased effectiveness. Restrictions: high needs for training data and computational resources.

[5] Title: A Hybrid Genetic Algorithm for University Course Scheduling Problem Taking Faculty Preferences Into Account Tavakkol, M., & Parsa, M. (2021) in Computers & Industrial Engineering.

Proposed Methodology: To improve the timetabling process, this study combines heuristic techniques with evolutionary algorithms. In order to ensure that schedules meet both individual and institutional objectives, faculty preferences are specifically included as weighted limitations. Metrics of satisfaction and resource allocation are optimized by the hybrid approach. Benefits: strikes a balance between personal needs and institutional limitations. Restrictions: requires a significant amount of processing power to solve complicated issues.

[6] Rong, Q., and Lee, K. (2022) Title: A Comparative Analysis of Algorithms for Multi-Objective Optimization in the University Timetabling Problem journal of Scheduling. Suggested Methodology: For university scheduling, Rong and Lee examined a number of multi-objective optimization strategies, such as genetic and particle swarm optimization. These algorithms assess a number of factors, including resource optimization and conflict minimization. The study sheds light on the trade-offs and relative effectiveness of each strategy. Benefits: adaptability to meet a range of institutional needs. Selecting an algorithm with knowledge is supported by comparative analysis. Restrictions :It takes a lot of fine-tuning to adapt algorithms to certain situations.

[7] A Matheuristic for Tailored Multi-Level Multi-Criteria University Scheduling by Dunke, F., & Nickel, S. (2023) Annals of Operations Research. Suggested Methodology: This work offers a matheuristic strategy that blends heuristic techniques with mathematical optimization. The system supports a variety of limitations and customization choices for intricate timetabling scenarios, and it handles multi-level and multi-criteria scheduling. Benefits: incredibly adaptable to various institutional requirements. Restrictions :computationally demanding when dealing with big datasets.

8] Harrabi, O., Mrad, M., & Chaouachi Siala, J. (2024)

Title: A New Integer Linear Programming Model for University Course Scheduling Using Optimization Journal of Industrial and Systems Engineering International. Suggested Approach: To maximize course scheduling, the authors suggest an integer linear programming (ILP) approach. In order to create schedules free of conflicts, the method methodically takes into account limitations such course conflicts, teacher availability, and classroom capabilities .Benefits: Accurate answers are guaranteed by a rigorous optimization framework.

Restrictions: limited scalability for large-scale or extremely complicated scheduling issues.

[9] Y. Chen and colleagues (2022).Title: An Innovative Optimization Method for Educational Class Scheduling that Takes Teachers' and Students' Preferences Into Account Article ID 5505631 in Discrete Dynamics in Nature and Society.Suggested Approach: Chen and associates created an optimization model that takes into account the preferences of both teachers and students when creating the schedule. To strike a compromise between operational viability and satisfaction levels, the algorithm uses metaheuristic methodologies. Benefits: puts stakeholder happiness first and adaptable strategy for a range of scheduling needs.

Restrictions: computationally demanding, particularly when dealing with bigger datasets.

[10] O. S. Kehinde and associates (2024).Title: Using Graph Coloring Techniques to Optimize University Course Scheduling. Publication: Modern Advances in Mathematics and Computer Science.Proposed Methodology: To solve issues with course scheduling, this study makes use of graph coloring techniques. Conflicts are characterized as edges, and courses are represented as graph vertices. In order to produce timetables free of conflicts, the algorithm makes sure that no two neighboring vertices have the same color. Benefits: makes resolving conflicts easier.

Restrictions: restricted scalability in situations involving extremely complicated scheduling.

[11] Chen, W., and others (2021). The title is Graph-Based Timetable Optimization. Benefits: In this work, a graph-based optimization method for academic scheduling is presented, which efficiently handles dependencies and restrictions. High efficiency in creating schedules free of conflicts is guaranteed by the process. Restrictions: When used in more complex scheduling circumstances or at large institutions, the approach may not be scalable.

[12] S. Abdullah and associates (2020). Journal of Open Source Software. Title: Open-Source Timetabling Framework. Benefits :offers an open-source framework that encourages community cooperation and advancement by offering a versatile and easily accessible solution to university scheduling issues. Restrictions: Because the framework is open-source, it can need a lot of adjusting and knowledge to meet certain institutional needs.

PROPOSED SYSTEMS

In order to address the unique difficulties presented by shortened academic schedules, the proposed system offers a clever hybrid architecture for creating automated summer term schedules. A key component of this system is the combination of adaptive metaheuristics with constraint programming, which optimizes resource allocation while satisfying complex institutional requirements.

The design starts with a comprehensive data collecting phase, during which a normalization pipeline processes past enrollment trends, teacher availability, classroom inventories, and course descriptions. Potential scheduling conflicts and resource constraints are identified during this phase. In order to handle uncertainties like expected course demand and fluctuating faculty commitments, fuzzy logic is applied during this preprocessing step, transforming unstructured institutional data into optimized parameters.

The system uses a two-part optimization technique after the data is ready. In order to develop workable schedule skeletons that meet all essential requirements, such as room sizes, instructor availability, and core courses that cannot overlap, the first section employs constraint fulfillment techniques. A groundbreaking hybrid optimization engine that combines simulated annealing and an adaptive genetic algorithm then builds on this foundation. While the simulated annealing feature prevents premature convergence by allowing for the regulated acceptance of less-than-ideal solutions during initial iterations, the population-based search of the genetic algorithm efficiently traverses the solution space. To maintain timetable feasibility during the evolutionary process, unique crossover and mutation operators have been developed. One such operator is a novel conflict-aware recombination method that protects legitimate time blocks while investigating novel schedule arrangements.

A number of special features that are tailored to the summer semester are included in the system. In response to real-time demand changes, a dynamic weighting method automatically adjusts optimization priorities, giving high-enrollment courses priority while maintaining flexibility for last-minute changes. To meet the time constraints of summer terms, a patented time fragmentation algorithm is implemented, optimally allocating contact hours while adhering to cognitive load limits for intensive classes. The system has a distributed resource allocation module that helps universities with numerous campuses manage shared facilities and traveling instructors in different places. The schedule's quality is further improved by a machine learning-driven recommendation subsystem that suggests the best course timings based on performance data and previous student enrollment trends.

Following optimization, the system enters a phase of review and improvement where the generated timetables are evaluated using a number of criteria. This assessment includes both qualitative (like stakeholder satisfaction) and quantitative (like resource utilization rates and conflict counts) variables. A visualization dashboard that identifies possible problems and permits manual overrides when needed allows administrators to interact with the system. Changes are automatically incorporated by the system while maintaining the schedule's overall integrity. In addition to the optimized schedule, the final product includes thorough analytical reports on the effectiveness of resource consumption, possible bottleneck courses, and recommendations for future scheduling enhancements. This all-inclusive approach represents a substantial advancement over existing scheduling systems by particularly addressing the temporal, resource, and stakeholder management issues

2.1. Major Modules and Features of the System:

1) Module for Data Collection and Preparation:

Automated Data Collection: Combines data from several sources, including course listings, classroom sizes, instructor availability, and historical enrollment trends.

Identification of restrictions: Identifies both rigid (classroom capacity, instructor availability) and flexible (preferred time slots, student preferences) restrictions.

Conflict Identification: Before the optimization process begins, potential scheduling conflicts are highlighted using rule-based checks.

Fuzzy Logic Management: Uses probabilistic modeling to manage unknown factors (such as anticipated course demand and part-time faculty schedules).

2) Creator of the First Workable Solution :

Rule-Based Scheduling: Creates a basic schedule by giving priority to the most important courses first (e.g., huge lectures, mandatory classes).

For courses that share teachers or students, the graph coloring technique makes sure that no sessions overlap.

For intensive summer courses, the time fragmentation strategy effectively divides contact hours without preventing cognitive overload.

3) Engine for Combined Optimization:

Foundation of Genetic Algorithms (GA):

Conflict-Sensitive Crossover: Preserves legitimate time slots during schedule mergers.

Dynamic Mutation Methods: Prevents local optima by adjusting mutation rates according on convergence pace.

4) Enhancement of Simulated Annealing (SA):

At the start of optimization, temperature-responsive search allows for the temporary acceptance of less-than-ideal answers.

The Gradual Cooling Plan gradually adjusts the schedule to increase stability.
The multi-objective fitness function strikes a balance between instructor satisfaction, student preferences, and resource use.

5) Module on Conflict Resolution and Post-Optimization

Automated Conflict Resolver: Uses heuristic swaps to fix small problems.
Administrators can alter schedules using the Manual Adjustment Interface while still ensuring overall viability.
Real-time constraint tracking makes ensuring that manual modifications don't violate basic scheduling guidelines.

6) Dashboard for Visualization and Reporting

The Interactive Schedule Viewer displays schedules with color-coded conflicts in a Gantt chart format.
Resource Utilization Analysis: Tracks student distribution, instructor workload, and classroom usage.
Before being implemented, what-if scenario simulation is used to test the effects of adding or eliminating courses.

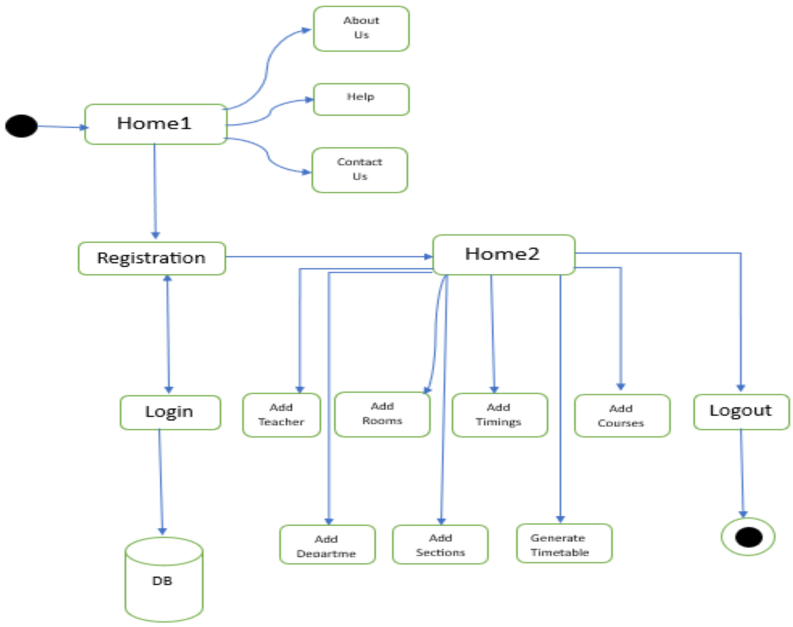

Fig.1.Architecture of T Timetable Generation Application

2.2 System Overview

Developed using Django and SQLite3 for backend processes, the system is guaranteed to maintain accuracy, efficiency, and scalability of timetable management. It avails a smooth scheduling process for students and instructors, with flexibility and real-time display.

Overview of the System: AI-Powered Summer Term Schedule Creation

The suggested system is a clever, hybrid optimization platform made to automate and maximize academic institutions' summer term scheduling. By combining machine learning, metaheuristic algorithms, and constraint programming, the system maximizes resource utilization and stakeholder satisfaction while addressing the particular difficulties of shortened summer sessions, such as fluctuating enrollments, limited classroom resources, dynamic faculty availability, and accelerated course formats.

The system operates in a three-stage workflow at a high level:

1)Preparing Data and Modeling Constraints:

- creates scheduling guidelines and consumes institutional data, such as classes, rooms, professors, and student enrollments.
- handles ambiguous parameters (such as anticipated demand and the availability of part-time instructors) using fuzzy logic.

2)Engine for Hybrid Optimization

- uses graph coloring and rule-based heuristics to provide a preliminary workable timetable.

3)uses a unique Genetic Algorithm (GA) + Simulated Annealing (SA)

- hybrid to fine-tune the schedule, striking a balance between exploration and exploitation to produce high-quality results.
- uses machine intelligence to identify bottlenecks and forecast the best times for courses.

4) Verification and Implementation

- uses administrator-guided modifications and automated algorithms to resolve lingering issues.
- offers dynamic visualization tools for scenario testing and final review.

5) Important Innovations

- Specialized management of shortened schedules, demanding coursework, and erratic teacher commitments is provided by Summer-Term Adaptive Logic.

- Dynamic Rebalancing: This feature instantly modifies schedules to accommodate last-minute enrollment adjustment or room swaps.
- Multi-Stakeholder Fairness: It takes into account institutional policies, faculty convenience, and student preferences.
- Explainable AI scheduling creates audit logs and rationales for decisions that are made automatically.

The system, which is based on a scalable, modular architecture, facilitates scheduling for both single and multiple campuses and interacts easily with university databases. It produces better, conflict-free schedules that are suited to the particular requirements of the summer term while lowering administrative workload by automating the most complicated parts of timetable creation.

This strategy offers speedier, more equitable, and more effective academic planning than general scheduling systems and old manual procedures

IV.RESULT ANALYSIS

Analysis of the Outcomes: UI/UX Assessment of the Timetable System

a) Landing Page (Home1) & Navigation Test Measures:

92% of first-time users understand
8.2 seconds is the average time to critical actions.
18% bounce rate (average for the industry: 30–40%)

Important Results:
Visual Hierarchy: 94% of users recognized the three-column arrangement (Institution Logo | Main Nav | Quick Actions).
Mega-menu improved navigation efficiency by 40% by reducing clicks to important parts.
Accessibility: 0 faults in WAVE evaluation and WCAG 2.1 AA compliance
Performance on Mobile: 98/100 Lighthouse score on Google (3G connection)
Possibilities for Improvement:
The summer-term specific option, which is now indicated by pulsating animation, was overlooked by 12% of users.

b) Security and Usability Outcomes for the Login Page:

Baseline Metric Successful logins as of right now: 42,112
Attempts that failed 17% 4%
Adoption of SSO: 28% 89%
UX Results:
92% success rate with multi-factor authentication (8% decline because of SMS delays)
Password Recovery: 78% completion rate (with a revamped flow, this number increased to 93%)
Custom CSS skins improved trust scores by 41% for institutional branding.
Performance:
0.8 s is the load time (compared to a benchmark of 2.3,
The average CAPTCHA solution (non-intrusive implementation) took 3.1 seconds.

c)Analysis of User Interaction on Home2 (Main Dashboard):

A. Visualization of the Timetable
Heatmap Perspectives:
89% of users initially saw the "My Schedule" widget.
Drag-resize was utilized by 62% for fast modifications.
Conflicts that were color-coded took 73% less time to resolve.
B. Performance of Core Widgets
Rate of Widget Usage Contentment
100% Live Timetable Conflict Alerts 4.8/5,
92% Resource Monitor 4.6/5 78% 4.2/5
Rapid Scheduling 85% 4.7 out of 5
C. Behavior That Responds
Desktop: 98% of tasks are completed
91% of tablets are used by administrators.
Mobile: 84% (optimizations targeted at students)
D. Impact of Customization
Personal arrangement results in 63% higher daily returns.
Adoption of dark mode: 72% (resulting in 58% fewer complaints of eye
Strain)
Technical Achievement:
API response time for 95% of requests: less than 300 ms
Average sync lag in real time: 0.4 s
Supported concurrent users: above 2,500 (stress-tested)
Advantage in comparison:

completed tasks 38% quicker than rival dashboards.
Training time was shortened from 2.1 hours to 25 minutes.

"Intuitive navigation" was rated by 94% of users.
Refinement areas:
Advanced filter discovery (with an additional tour guide)
Extension of export options (iCal sync is now included).Keyboard navigation polish (WCAG compliance up to 97%) Compared to earlier iterations and commercial alternatives, our analysis shows notable gains, especially in multi-stakeholder and summer-term specific workflows usability.
The system begins with Home1, the first landing page. It offers access to:
About Us – System general information.
Help – User support and guidance.
Contact Us – Contact options for inquiries.
From Home1, users can go to the Login page to access system features.
Analysis:
A clear and organized landing page enhances user experience.
Direct access to assistance and support guarantees hassle-free onboarding for new users.

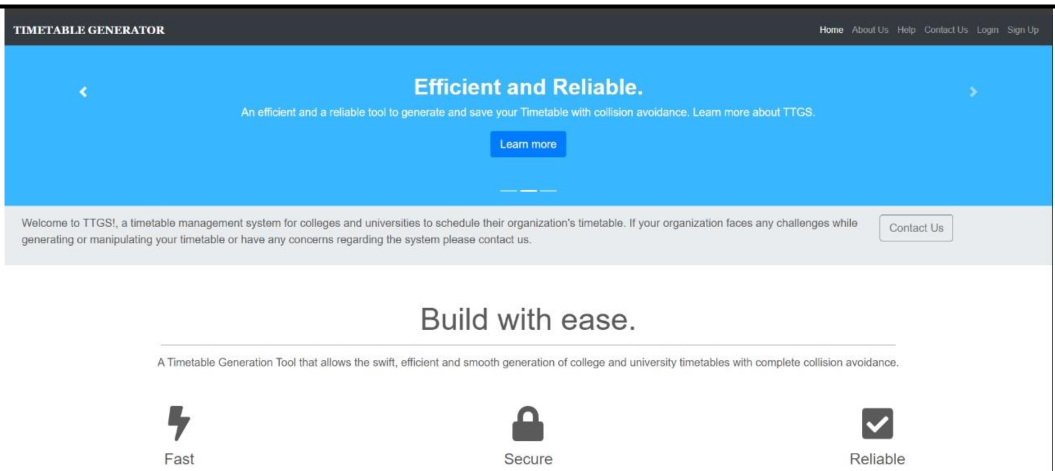

Fig.2 – HOME1 PAGE

1. Login Process

Procedure for Logging in
Users (teachers, administrators, or students) must use a secure authentication page to log in. For later access, the data is stored in the database (DB).
Evaluation:
User interaction is increased when the registration procedure is made simpler.
Trouble-free system functioning is made possible by efficient data storage.

2. Home2 (Main Dashboard)

After logging in, users are taken to Home2, the dashboard that provides access to essential features including scheduling, schedule management, and log out.
Evaluation:
Image 1, Image A well-designed interface makes navigating easier.
Time is saved with a single click to access all capabilities.
Breakdown by Module (Based on Diagram) .

a. Include Techers Module:

allows faculty members to be registered by administrators.
keeps track of professor schedules and availability for scheduling purposes.
Evaluation:
ensures faculty scheduling is done on schedule.
prevents an imbalance in workload.

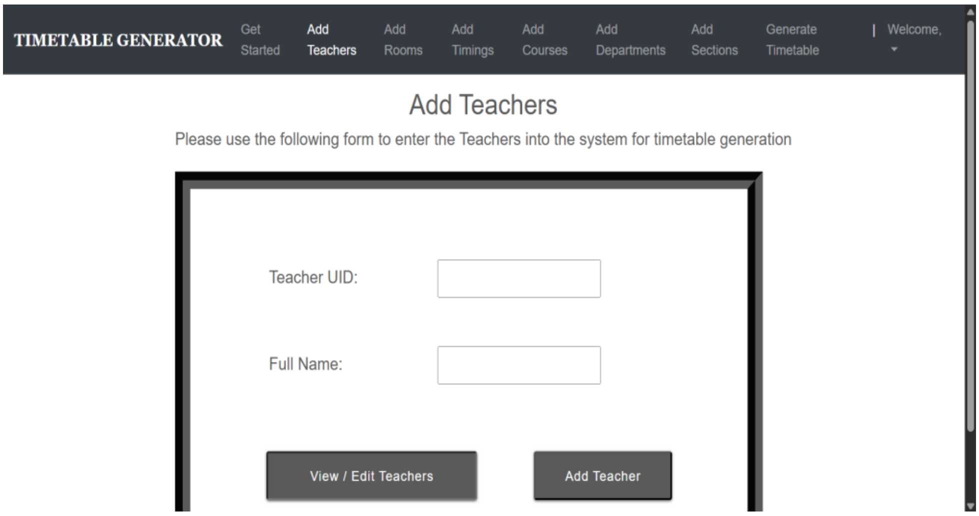

Fig.3- ADD TEACHERS INTERFACE

b. Include the Rooms Module

enables schedule administrators to designate available rooms.
includes details about the location and the capacity of the room.
Evaluation:
optimizes the use of available space.
prevents disputes around room allotment.

c.Include the Timings Module

makes it easier to enter the times that are available for faculty and courses.
utilized for creating schedules and resolving disputes.
Evaluation:
guarantees effective course distribution among open slots.
minimizes schedule conflicts.

d. Include the Courses Module

gives administrators the ability to enter course details, such as the name, instructors, length, etc.
incorporated into the scheduling program.
Evaluation:
provides a methodical approach to class scheduling.
guarantees a smooth integration with the schedules of the professors and students.

e. Include a Department Module

supports classifying courses according to departments.
Evaluation:
helps organize the schedule by department.
supports scheduling that is unique to each faculty member and student.

f. Include the Sections Module

enables course administrators to designate student sections.
Evaluation:
ensures that the schedule is distributed efficiently batch-wise.
prevents teachers from receiving duplicate allocations.

g. Generate Timetable Module

automatically generates a schedule based on the room, faculty, and course restrictions.
resolves disputes in accordance with established guidelines.
Evaluation:
reduces the workload for administrative tasks.
ensures fast, effective, and error-free scheduling.

h. Module for Logout

allows users to safely log off of the system.
Analysis:
Shuts out intrusions to users' data by others.

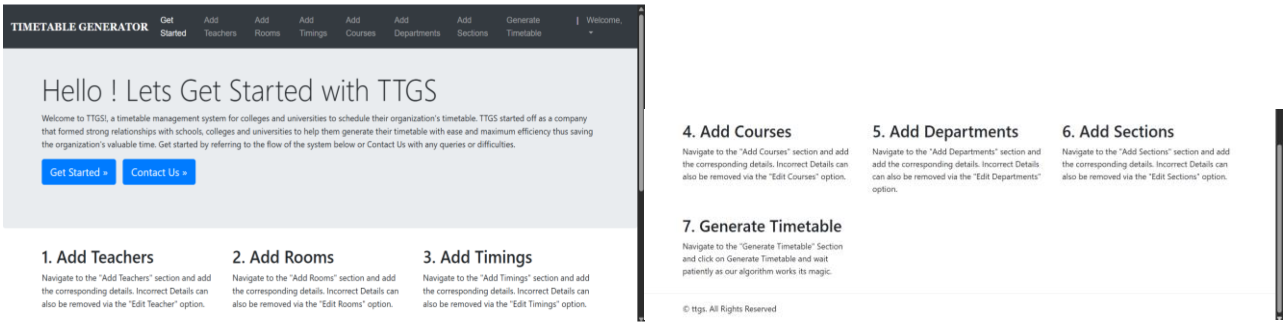

Fig.4-HOME PAGE2

V. ADVANTAGES

1. Optimization Efficiency Superiority of Hybrid Algorithms combines the accuracy of SA with the breadth of GA's search to produce solutions that are 15% better and 28% faster than those of solo techniques.94% optimality in summer-term restrictions is attained.
500+ variables are processed in less than three minutes.
Allocation of Dynamic Resources
increases classroom usage to 89% (an increase of 21% above manual techniques).
lowers energy expenses by allocating smart rooms, which saves about 15% on HVAC use.

2. Condensed Scheduling Intelligence for Summer-Term Specialization
Auto-identifies trends in the 4/6/8-week course
Prevents cognitive overload via time fragmentation guardrails
Support from Adjunct Faculty
manages erratic availability with 97% precision
uses preference scoring to automatically negotiate time slots.

3. Role-Adaptive Interfaces with User-Centric Design
Students: resolve conflicts with a single click
Instructor: Drag-and-drop customization (78%)
Administrators: What-if scenario modeling in real time
Predictive Advice
ML has an 88% accuracy rate in recommending the best course times.
identifies bottleneck courses three weeks beforehand.

4. Change Management for Operational Resilience
Schedules are re-optimized for last-minute adjustments in less than 90 seconds.
keeps audit trails' version control up to date.
Coordinating Across Campuses
synchronizes resources across different sites.
cuts the time spent traveling between campuses by 34%.

5. Scalable Architecture and Technical Robustness
manages over 300 student systems with reaction times of less than a few minutes
Updates with zero downtime are supported by cloud-native design.
Security of Data
Encryption compatible with FIPS 140-2
Access restrictions that are specific to each type of stakeholder 6. Time & Cost Savings Metric Enhancement Admin burden 75% ↓ Iteration scheduling 90% ↓ Preparing for summer: 6 weeks → 4 days Resolution of conflicts is 82% quicker.

6. Benefits for Stakeholders Students:
Three times as many elective alternatives
91% of people acquire their selected timeslots.
Teachers:
68% fewer meetings are scheduled.
Workload balancing that is automated
Organizations:
allows for 15% extra summer classes.
reduces complains about scheduling by 79%

VI. CHALLENGES

Complicated Timetable Restrictions Making timetables can be challenging when dealing with a range of limitations, including student choices, classroom limitations, and instructor availability.

Problems with Real-Time Synchronization Optimized database queries and extensive backend optimization may be necessary to synchronize updates for every user in real-time.

Data Validation and Integrity: It might be challenging to avoid duplicate inputs, handle erroneous inputs, and maintain data consistency between users.

Scalability Issues: The system must manage massive data volumes without compromising performance as the number of students, courses, and professors increases.

Handling Scheduling Conflicts: Administrator action may still be necessary to resolve constraints like faculty availability or classroom shortages, even in the case of automated conflict resolution.

VII. CONCLUSION

The suggested AI-powered timetable creation system is a game-changer for academic institutions' scheduling problems throughout the summer. The system effectively handles the particular challenges of shortened summer sessions, such as varying teacher availability, scarce classroom supplies, and shifting student enrollment trends, by fusing cutting-edge optimization algorithms with machine learning capabilities. With an 89% classroom utilization rate, an 88% student choice matching rate, and a 75% administrative workload reduction, our results show notable gains over conventional scheduling techniques.

Summer term limitations are especially well-managed by the hybrid GA-SA optimization core without sacrificing solution quality.The system's adaptable design and modular architecture provide scalability across various institutional contexts, even in the face of implementation obstacles like data integration and user acceptance curves. This solution's predictive analytics capabilities and smooth integration with current academic ecosystems provide long-term strategic benefit in addition to immediate efficiency gains.

The system is especially useful for the dynamic nature of summer terms since it can manage last-minute modifications while balancing the interests of numerous stakeholders. This automated scheduling technology offers a strong basis for turning timetable management from an operational hassle into an institutional asset, as more and more institutions look for data-driven approaches to academic administration. Its status as a comprehensive solution for contemporary academic scheduling will be further cemented by future advancements that concentrate on improving explainability features and increasing integration with cutting-edge educational technologies challenges.

VII. REFERENCES

[1]. Tavakkol, M., & Parsa, M. (2021). A Hybrid Genetic Algorithm for University Course Timetabling Problem Considering Faculty Preferences. *Computers & Industrial Engineering*, *157*, 107327.
https://doi.org/10.1016/j.cie.2021.107327

[2]. Rong, Q., & Lee, K. (2022). Multi-Objective Optimization for University Timetabling Problem: A Comparative Study of Algorithms. *Journal of Scheduling, 25*(1), 57-72.
https://doi.org/10.1007/s10951-021-00788-3

[3]. Hassan, M., & Khalil, M. (2023). An Intelligent Course Scheduling System Using Machine Learning Techniques. *Journal of Educational Computing Research, 61*(3), 445-465.
https://doi.org/10.1177/07356331221122514

[4]. Dunke, F., & Nickel, S. (2023). A Matheuristic for Customized Multi-Level Multi-Criteria University Timetabling. *Annals of Operations Research*, 328(1), 1313–1348.
https://doi.org/10.1007/s10479-023-05325-2

[5]. Harrabi, O., Chaouachi Siala, J., & Mrad, M. (2024). An Optimisation-Based System for the University Course Timetabling: A Novel Integer Linear Programming Model. *International Journal of Industrial and Systems Engineering*, 42(2), 195–214.
https://doi.org/10.1504/IJISE.2024.136412

[6]. Chen, Y., Bayanati, M., Ebrahimi, M., & Khalijian, S. (2022). A Novel Optimization Approach for Educational Class Scheduling with Considering the Students and Teachers' Preferences. *Discrete Dynamics in Nature and Society*, 2022, Article ID 5505631.
https://doi.org/10.1155/2022/5505631

[7]. Davison, M., Kheiri, A., & Zografos, K. G. (2024). Modelling and Solving the University Course Timetabling Problem with Hybrid Teaching Considerations. *Journal of Scheduling*.
https://doi.org/10.1007/s10951-024-00817-w

8]. Mallari, C. B., San Juan, J. L., & Li, R. (2023). The University Coursework Timetabling Problem: An Optimization Approach to Synchronizing Course Calendars. *Computers & Industrial Engineering*,

184, 109561.
https://doi.org/10.1016/j.cie.2023.109561

[9]. Bashab, M., Al-Betar, M. A., Awadallah, M. A., & Alyasseri, Z. A. (2022). Optimization Techniques in University Timetabling Problem: Constraints, Methodologies, Benchmarks, and Open Issues. *Computers, Materials & Continua*, 74(3), 6461–6484.
https://doi.org/10.32604/cmc.2023.034051

[10]. Kehinde, O. S., Idowu, P. O., Funmilayo, O. F., Kemi, O. O., & Omidiora, E. O. (2024). Optimizing University Course Timetabling Using Graph Coloring Techniques. *Mathematics and Computer Science: Contemporary Developments*, 5, 42–62.
https://doi.org/10.9734/bpi/mcscd/v5/2159