A
Project Report
on

DYNAMIC TIMETABLE GENERATOR USING GENETIC ALGORITHM

Submitted to

Sant Gadge Baba Amravati University, Amravati

Submitted in partial fulfilment of

the requirements for the Degree of

Bachelor of Engineering in

Computer Science and Engineering

Submitted by

Kunjan Katore
(PRN: 213120432)
Rasika Lahane
(PRN: 213120346)

Punam Motewar
(PRN:213120324)
Chaitali Nakhate
(PRN: 213120144)

Under the Guidance of
Dr. P. K. Bharne
Assistant Professor, CSE Department

Department of Computer Science and Engineering
Shri Sant Gajanan Maharaj College of Engineering,
Shegaon – 444 203 (M.S.)

Session 2024-2025

SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING,
SHEGAON – 444 203 (M.S.)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that **Ms. Kunjan Katore, Ms. Punam Motewar, Ms. Rasika Lahane and Ms. Chaitali Nakhate** students of final year Bachelor of Engineering in the academic year 2024-25 of Computer Science and Engineering Department of this institute have completed the project work entitled "**Dynamic Timetable Generator Using Genetic Algorithm**" and submitted a satisfactory work in this report. Hence recommended for the partial fulfilment of degree of Bachelor of Engineering in Computer Science and Engineering.

Dr. P. K. Bharne
Project Guide

Dr. J. M. Patil
Head of Department

Dr. S. B. Somani
Principal
SSGMCE, Shegaon

SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING,
SHEGAON – 444 203 (M.S.)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that **Ms. Kunjan Katore, Ms. Punam Motewar, Ms. Rasika Lahane and Ms. Chaitali Nakhate** students of final year Bachelor of Engineering in the academic year 2024-25 of Computer Science and Engineering Department of this institute have completed the project work entitled "**Dynamic Timetable Generator Using Genetic Algorithm**" and submitted a satisfactory work in this report. Hence recommended for the partial fulfillment of degree of Bachelor of Engineering in Computer Science and Engineering.

Internal Examiner
Dr. P.K. Bharne
Name and Signature
Date: 9/5/25

External Examiner
A. D. Shah
Name and Signature
Date: 9/5/25

Acknowledgement

It is our utmost duty and desire to express gratitude to various people who have rendered valuable guidance during our project work. We would have never succeeded in completing our task without the cooperation, encouragement and help provided to us by then. There are a number of people who deserve recognition for their unwavering support and guidance throughout this report.

We are highly indebted to our guide **Dr. P. K. Bharne** for his guidance and constant supervision as well as for providing necessary information from time to time. We would like to take this opportunity to express our sincere thanks, for his esteemed guidance and encouragement. His suggestions broaden our vision and guided us to succeed in this work.

We are sincerely thankful to **Dr. J. M. Patil,** HOD, CSE Department, SSGMCE, Shegaon, and to **Dr. S B Somani,** Principal, SSGMCE, Shegaon who always has been kind to extend their support and help whenever needed.

We would like to thank all teaching and non-teaching staff of the department for their cooperation and help. Our deepest thank to our parents and friends who have consistently assisted us towards successful completion of our work.

Kunjan Katore (22BD310761)
Punam Motewar (22BD310778)
Rasika Lahane (22BD310781)
Chaitali Nakhate (22BD310741)

Contents

Abstract

Generating efficient timetables in academic institutions is a constraint-based, complex problem encompassing faculty availability, resource assignment, and class scheduling. Manual approaches tend to cause conflicts, inefficiencies, and rigidity problems that increase with institutional expansion. In response, we introduce a web-based automated timetable generator for our college with three roles: Admin, Teacher, and Student. Admins control data (students, teachers, subjects, classes, labs, branches) and create timetables. Teachers have access to schedules and provide feedback, and students view their course-specific timetable. A Genetic Algorithm (GA), based on natural selection, iteratively evolves timetables optimized in the system. One-point crossover is used in each generation to blend strong characteristics of parent timetables and mutation to add diversity, reducing conflicts and maximizing resource use. Developed using HTML, CSS, JSP (frontend), Java Servlets (backend), and MySQL (database), the system is dynamic, scalable, and responsive to real-time changes like faculty unavailability or rescheduling of classes. Utilizing AI optimization and web technologies, it offers an effective, flexible solution for institutional scheduling requirements. It successfully automates timetable generation using advanced algorithms, improving scheduling efficiency and accuracy. It provides a scalable, web-based solution adaptable to the evolving needs of educational institutions.

Keywords: *Genetic Algorithm, Timetable Automation, Java Servlet, Web Application, JSP, MySQL.*

List of Figures

List of Screenshot

CHAPTER 1

INTRODUCTION

INTRODUCTION

1.1 PREFACE

In the fast-changing educational environment of today, institutions are confronted with increasing complexity in the management of academic operations. Among these operations, generation of timetables is one of the most complex and time-consuming activities. From primary schools to universities, generation of effective and conflict-free timetables is crucial for the smooth functioning of academics. Institutions need to manage a large number of variables such as faculty availability, classroom allocation, course requirements, and student enrollment. The objective is to make sure that all resources are utilized in an optimal manner without compromising the integrity of the academic timetable. However, accomplishing this manually becomes more and more impractical as the number of courses, instructors, and constraints increases. This fact makes the implementation of automated systems necessary to manage this complexity in an intelligent and dynamic way [1][4].

The conventional method of timetable generation is for administrators to spend hours on end manually inputting data, checking availability against each other, and making adjustments to prevent clashes. Although this technique might be adequate in small institutions with few variables, it becomes impossible to handle as the institution expands. Manual systems are not only time-consuming but also extremely prone to errors and omissions, which can result in scheduling clashes, underutilization of resources, and administrative burnout [1][5-6]. The incorporation of computerized systems most importantly, those based on optimization software such as Genetic Algorithms has proven to be a feasible and effective option. These systems have the potential to greatly automate the process by implementing hard and soft constraints, responding to institutional guidelines, and producing practicable timetables that enhance overall efficiency [8].

1.2 MOTIVATION

The need to develop an automated system for generating timetables stems from the urgent necessity to replace inefficient manual methods with intelligent and responsive ones. Schools and colleges increasingly face the dual pressure of increasing academic activities and raising administrative effectiveness. With such an environment, a high-performance scheduling system no longer remains an extravagance but an absolute necessity [6][8]. Manual scheduling in the traditional sense is prone to errors such as double-booking rooms, duplicate assignment of faculty, or ignoring institutional policies. In addition, manual scheduling is static and thus less than optimal in handling short-notice reassignments such as faculty unavailability or sudden closure of rooms. A computer system, however, can dynamically adjust to such changes, rapidly recalculate timetables, and maintain interruptions to a minimum. This is particularly important in today's high-speed educational institutions where speed and accuracy are paramount.

The second compelling reason is the potential for resource optimization. Teaching resources such as classrooms, laboratory time, and instructor time are finite and have to be used effectively. An improperly designed timetable can create space underutilization, workload overload, and dissatisfied students. In contrast, an optimized schedule ensures equitable distribution of teaching loads, minimizes waste times, and maximizes total academic productivity. Further, such a system can minimize soft constraints such as faculty preferences or preferred time periods, optimizing satisfaction and engagement. This automated process reduces administrative burden and allows the staff to invest time in upper-level activities. With the application of sophisticated computational techniques, academic institutions can embrace a more structured, efficient, and responsive academic schedule model [6][10].

1.3 PROBLEM STATEMENT

The current manual process of schedule generation in institutions is inefficient, time-consuming, and error-prone. Administrators often have a hard time ensuring that all constraints, such as faculty availability, room assignments, and course requirements, are well addressed. This results in conflicting schedules, such as double-booking the rooms or conflicting the faculty appointments, which disrupt the educational schedule. Furthermore, the manual systems lack flexibility to simply react to spontaneous changes, such as faculty members' availability or reassignment of rooms. The current process, therefore, cannot be scaled up and does not optimize the utilization of resources, causing added administrative weight and impairing operational efficiency.

1.4 AIM & OBJECTIVES

AIM: The aim of the project is to develop a web-based timetable generator for college that efficiently schedules classes while accommodating various constraints and user roles using a genetic algorithm.

To fulfill the aim of the project, the following are the objectives:

a) To Study the Conventional Methods of Generating Timetable.

b) To develop a cross-platform application using HTML, CSS, JSP for the frontend.

c) To develop backend using JavaScript, Java, MySQL.

d) To compare the algorithm from various research paper and finding the best one to fit according to the project.

1.5 ORGANIZATION OF PROJECT

Chapter 1: Introduction

This chapter introduces the project's aim to simplify timetable creation using a user-friendly web application. Built on the Eclipse framework, it leverages genetic algorithms to generate optimized, conflict-free schedules, improving resource utilization and efficiency.

Chapter 2: Literature Review

This section reviews existing research on timetable optimization techniques, highlighting traditional limitations and the need for automation. It examines heuristic, AI, and machine learning methods, identifying gaps that guide the project's methodology.

Chapter 3: Analysis of Project Requirements

This chapter analyzes functional and non-functional requirements, addressing stakeholder needs, user roles, and data constraints. It outlines key features and defines performance indicators essential for evaluating system effectiveness.

Chapter 4: Architecture and Design

Here, the system architecture and design principles are detailed, covering database structures, API integrations, and UI components. The focus is on scalability, maintainability, and providing a seamless user experience.

Chapter 5: Implementation

This chapter outlines the development process, including coding, module-wise testing, and system integration. It also discusses debugging, performance optimization, and plans for future feature enhancements.

Chapter 6: Result and Discussion

This section presents the outcomes of the project, comparing automated schedules with manual ones. It evaluates the system based on resource usage, conflict resolution, and user feedback to assess effectiveness and usability.

Chapter 7: Conclusion

The final chapter summarizes the project's objectives, methods, and results while suggesting future improvements. It explores integrating predictive analytics and the potential to expand the system for corporate or healthcare scheduling.

CHAPTER 2

LITERATURE
REVIEW

LITERATURE REVIEW

A literature review involves systematically reviewing existing research on a specific topic, providing a critical summary, identifying gaps, and offering insights for future studies. It serves to contextualize, evaluate, and synthesize the current state of knowledge in a particular field.

"A Novel Genetic Algorithm-Based Timetable Generator for Optimized University Timetable Solution" by Ali Hasan Khan and Talha Imtiaz (2024) introduces a sophisticated method of solving the intricate university timetable scheduling problem through genetic algorithms (GA). The research highlights the efficiency of GA in optimizing timetable generation by minimizing scheduling conflicts and enhancing resource allocation. Using one-point crossover and mutation operations, the algorithm repeatedly improves the timetables produced during successive generations so that the output is within specified constraints and offers an optimal scheduling solution. The research points out that genetic algorithms, based on the mechanisms of natural selection and evolution, are particularly appropriate for solving combinatorial optimization problems such as timetable scheduling. The efficiency of the system lies in its ability to search through a large solution space and move towards an optimal schedule with fewer conflicts. In contrast to conventional scheduling systems that depend on manual tuning or heuristic methods, the GA-based system schedules automatically while taking into account important constraints like classroom availability, instructor preferences, and course prerequisites. Though its merit, the paper also pinpoints a crucial flaw in the strategy: heavy dependency on predetermined constraints, thereby limiting the flexibility of the system in adapting to changes in real-time scheduling. The academic environment itself is dynamic, with most requiring last-minute changes due to teacher unavailability, student class list fluctuations, or administrative corrections. The GA-based system, although good in a static setting, finds it challenging to deal with such real-time changes. The inflexibility comes from the nature of genetic algorithms to use several rounds of iterations in order to improve solutions, and hence, the incorporation of quick adjustments would not be feasible without the need to restart the overall optimization process. The inflexibility lowers the usability of the system in real-life academic environments where flexibility is paramount. The study indicates that any future development should center on incorporating adaptive mechanisms that enable real-time adjustment without undermining the efficiency of the scheduling process. Potential enhancements may include hybrid optimization methods that integrate GA with reinforcement learning or heuristic-based methods to increase flexibility. Moreover, incorporating a feedback loop that allows the system to dynamically adapt constraints according to changing requirements can greatly enhance its practical applicability. The research concludes that although genetic algorithms provide a very promising solution for timetable optimization, more work needs to be done to make them more responsive and integrate them easily into dynamic educational settings.[1]

"Automatic Time Table Generation Using Genetic Algorithm" by Mrs. G. Maneesha et al. (2021) explores the implementation of genetic and heuristic algorithms to automate the academic timetable creation process. The study emphasizes the inefficiency of manual scheduling and advocates for an automated system that handles subject requirements, faculty availability, and workload distribution. Genetic algorithms improve the scheduling process by iteratively selecting optimal solutions using selection, crossover, and mutation operators. The research highlights the system's success in minimizing scheduling complexity, class clashes, and resource misallocation. Heuristic rules further enhance decision-making by prioritizing key constraints. However, the study identifies a major limitation in the system's inability to handle last-minute changes or dynamic adjustments. Any change requires restarting the entire GA process, which is computationally expensive and impractical. To solve this, the study recommends developing dynamic scheduling models or hybrid systems combining GA with machine learning. Interactive user interfaces and adaptive heuristics could improve flexibility. The paper concludes that while GAs enhance scheduling efficiency, adaptability and responsiveness are essential for real-world applications.[2]

"Design and Implementation of Time Table Generator" by Varsha S et al. (2022) presents a software-based approach for automating timetable generation in academic settings. The system minimizes manual errors, reduces administrative workload, and ensures transparency in faculty and classroom allocation. Key constraints like teacher availability, course requirements, and institutional policies are addressed efficiently. The algorithmic method enables faster timetable creation compared to traditional methods. The user-friendly interface allows easy input and modification of constraints.

Although effective for small institutions, the paper notes scalability issues in larger institutions due to increasing complexity and computational demands. The system may underperform when handling numerous departments or dynamic scheduling needs. The study suggests enhancements through advanced optimization methods like GAs, heuristics, or machine learning. Cloud computing is also proposed to manage large-scale workloads. Adaptive algorithms capable of real-time updates are recommended to respond to sudden changes like staff absences. Overall, the study underlines the benefits of automation in scheduling while highlighting the need for improved scalability and flexibility.[3]

"Automatic Timetable Generation System" by Rajshri Firke et al. (2023) investigates a scheduling solution based on the Rapid Application Development (RAD) model to replace manual timetabling in schools. The RAD model enables fast feedback integration and iterative development for timetable generation. This increases responsiveness and reduces manual scheduling errors. The system accommodates constraints like teacher availability, room allocation, and subject requirements through a simple user interface. Automation saves time and allows staff to focus on educational quality. However, a key limitation is the absence of sophisticated optimization algorithms, which restricts performance in complex scheduling scenarios. The paper states that the current RAD model is insufficient for large institutions with many scheduling variables. Suggested improvements include integrating GAs, heuristics, or AI to resolve complex conflicts and optimize scheduling. Machine learning could improve performance by learning from historical data, and constraint satisfaction techniques may reduce overlaps. A hybrid model combining RAD and optimization techniques would boost flexibility and scalability. The study concludes that automation works well but requires optimization methods for handling complexity.[4]

"Automated Timetable Generation Using Genetic Algorithm" by Shraddha Thakare et al. (2020) provides a genetic algorithm-based method to tackle inefficiencies in manual scheduling. GAs mimic natural selection using selection, crossover, and mutation to refine schedules based on constraints such as room availability, workload balance, and clash prevention. The study showcases how GAs enhance efficiency, diversity in solution space, and conflict resolution in academic timetabling. Despite these advantages, the paper identifies a limitation in traditional GAs when dealing with highly constrained or large-scale problems. Simple genetic operators may not find optimal solutions efficiently under complex conditions. To address this, the authors propose hybrid methods such as combining GAs with simulated annealing, PSO, or constraint satisfaction algorithms. Enhanced mutation strategies like adaptive mutation rates may improve convergence and performance. Future directions include dynamic scheduling and machine learning to improve flexibility and responsiveness. Overall, the study supports the use of GAs in timetable generation while advocating enhancements for real-world complexity.[5]

"Automated Timetable Generator" by Prof. Er. Shabina Sayed Ansari Ahmed et al. (2015) explores an evolutionary algorithm-based system that optimizes both hard and soft constraints in timetable generation. The research identifies the benefits of automation in reducing manual efforts and improving efficiency. By utilizing evolutionary computation, the system optimizes scheduling based on faculty, room, and subject constraints. However, it faces significant issues with real-time adaptability. The inability to handle unexpected changes like faculty absences or urgent rescheduling reduces the system's practicality in dynamic academic settings. The study recommends enhancements such as incorporating machine learning for predictive adjustments, dynamic constraint handling, and feedback loops for schedule updates. These improvements would enable better real-time responsiveness and flexibility. The authors emphasize that while evolutionary methods offer powerful tools for timetable automation, future research should focus on adaptability and real-world application. Hybrid strategies and AI-driven models could help the system better serve evolving educational environments and complex scheduling scenarios.[6]

"Smart Timetable System Using Machine Learning and Artificial Intelligence" by Suraj Nagtilak et al. (2023) presents an AI-based method for generating academic timetables. The study demonstrates how AI and ML enhance scheduling efficiency by considering multiple constraints such as staff availability, room allocation, and student course enrollments. These AI-powered systems dynamically optimize resource utilization and reduce scheduling conflicts. By leveraging historical data, they identify patterns to improve decision-making and scheduling strategies. However, the authors highlight a significant limitation dependence on past data for training ML models which affects institutions with limited datasets. In such cases, the model may struggle to generalize and fail to produce optimal schedules. To address this, the paper suggests reinforcement learning and real-time data integration as alternative approaches. Reinforcement learning enables the system to learn from real-time feedback and gradually improve its performance. The integration of real-time data such as faculty availability and student enrolment enhances adaptability. The paper also notes that AI methods must still improve in terms of scalability and responsiveness. It emphasizes the potential of hybrid optimization models and intelligent systems for future research. These approaches aim to improve scheduling quality in dynamic academic environments. The study offers a comprehensive overview of AI-driven timetable systems. It acknowledges their promise while encouraging continuous development.[7]

"Automatic Timetable Generator" by Rathod, P. P., Kamlesh K. Lodhiya et al. (2016) focuses on building an automated system to replace manual scheduling in academic institutions. The system addresses issues like faculty clashes, classroom assignment problems, and inefficient resource use. It adopts a rule-based approach that considers variables such as faculty preferences, course durations, and room availability. The interface allows users to input data manually, which the system then processes to generate a conflict-free timetable. The main objective is to reduce manual workload and streamline scheduling processes. Hard constraints like non-overlapping sessions are strictly enforced, while soft constraints aim for scheduling balance. The system minimizes faculty idle time and student overload. It is most effective in small-to-medium institutions with fewer departments. The simplicity of its user interface is a key advantage. However, limitations arise with increasing complexity, such as managing cross-departmental schedules. The algorithm's fixed nature restricts dynamic adjustments, making real-time responsiveness a challenge. The authors suggest future improvements like integrating genetic algorithms and feedback-based scheduling. These would allow the system to adapt better to changing conditions. The study contributes to academic automation and highlights room for growth through intelligent systems.[8]

"Automatic Timetable Generation System" by Deeksha et al. (2015) introduces an automated solution to replace inefficient and error-prone manual timetable generation. The proposed system employs constraint-satisfaction logic to build feasible schedules based on hard and soft constraints. Hard constraints include avoiding teacher overlaps and exceeding room capacity, while soft constraints improve user preferences like ideal teaching hours. Data such as faculty availability, subject credits, and classroom details are used as input. The system uses iterative refinement to resolve conflicts and optimize preferences. A key strength is its modular design, which allows different administrative functions like teacher assignment and room allocation to be handled independently. The system is especially suited for small to mid-sized institutions. It also provides user-friendly visual representations of schedules. However, the model's static logic limits adaptability to unexpected changes, like last-minute faculty absences. Performance drops in larger datasets or heavily constrained situations. The authors suggest enhancing the system with evolutionary algorithms such as genetic algorithms and simulated annealing. These could expand the solution space and improve adaptability. Real-time feedback and ML integration are also proposed for self-learning capabilities. The study provides a solid base for future intelligent and scalable scheduling systems.[9]

"Timetable Generation System" by Chowdhary et.al. (2015) presents an automated academic scheduling tool designed to replace traditional manual methods of timetable creation. As institutions grow and the complexity of scheduling increases, it becomes increasingly difficult to manually coordinate multiple variables such as course assignments, faculty availability, and room resources. This system aims to address these challenges by utilizing a rule-based approach to timetable generation, considering both hard constraints (such as preventing double bookings and ensuring no conflicts between faculty and room assignments) and soft preferences (such as minimizing idle time between classes). The system uses a sequential placement algorithm paired with a backtracking approach for conflict resolution, ensuring that a feasible timetable is generated efficiently. The simplicity and lightweight nature of the tool make it ideal for smaller educational institutions, where the scheduling needs are less complex. It allows for the creation of clear, user-friendly timetables that can be printed or digitized for easy access. However, the study highlights several limitations of the system, particularly its focus on generating the first feasible solution rather than an optimized one. As a result, timetables produced by the system may suffer from issues such as unbalanced workloads for faculty or unnecessary gaps between class periods, which may not be ideal from an operational perspective. While this approach meets basic scheduling needs, it lacks the ability to optimize timetables in the way that advanced techniques, such as genetic algorithms, might. The authors suggest that incorporating genetic algorithms could allow for more effective exploration of the solution space, offering better solutions that balance conflicting constraints more effectively. Additionally, the system’s inability to handle dynamic changes or real-time adjustments such as faculty absences or last-minute course additions—further limits its usability in a real-world context. To address this, the paper proposes integrating artificial intelligence (AI) to enable the system to learn from historical data and adapt to evolving scheduling needs. Such an AI-driven system could offer more flexibility and accuracy by adapting to changes without restarting the scheduling process. Moreover, the study emphasizes the importance of scalability, suggesting that as institutions grow larger and more complex, the system would need to accommodate greater volumes of data and more intricate constraints. To achieve this, the authors recommend the integration of cloud-based solutions for handling large datasets and more advanced algorithms that can improve computational efficiency and effectiveness. Real-time updates are also highlighted as a key requirement, with mobile and web-based solutions suggested for accessing and modifying timetables. These enhancements would not only increase the system's usability in larger institutions but also improve its ability to respond quickly to changes in a dynamic academic environment. Overall, while the system offers a valuable foundation for automating timetable generation, the study stresses the need for further improvements, particularly in optimization, scalability, and adaptability, to meet the increasingly complex and dynamic needs of modern academic scheduling. The authors conclude by suggesting that the future of academic scheduling will rely heavily on advanced optimization techniques and AI-driven solutions to improve both the efficiency and flexibility of timetable management in educational settings.[10]

CHAPTER 3

METHODOLOGY

METHODOLOGY

3.1 REQUIREMENT ANALYSIS

The first step involves gathering requirements from all relevant stakeholders, including administrators, teachers, and students. These are collected through interviews, surveys, and meetings. Key needs include room availability, faculty workload, and non-conflicting schedules. Functional requirements such as timetable generation, user authentication, and feedback collection are also recorded [4][7].

3.1.1 USE CASE DIAGRAM:

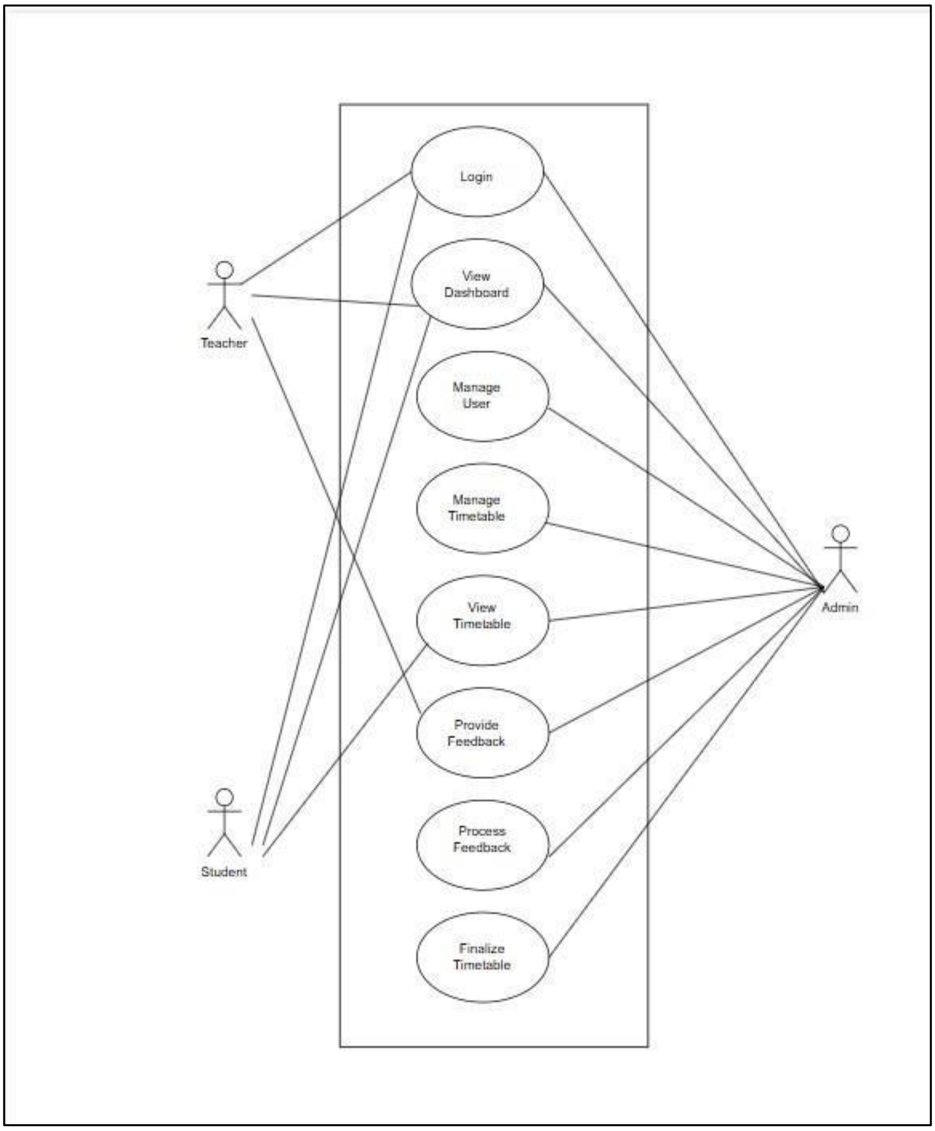


Figure 3.1: Use Case Diagram

The diagram is a Use Case Diagram of a Timetable Management System that indicates how various users interact with the functionalities of the system. There are three actors in it: Admin, Teacher, and Student. All the users first need to log in in order to use the system. Admin has maximum control, and it can add, edit users, timetables, process feedback, and finalize the schedule. Teachers can see the timetable, access the dashboard, and give feedback, whereas Students mainly see the timetable and give feedback. The system enables teachers and students to give feedback on the schedule, which is reviewed by the admin before the timetable is finalized. This controlled workflow supports a well-organized timetable administration process, striking a balance between administrative authorization and user feedback.

3.1.2 FLOWCHART:

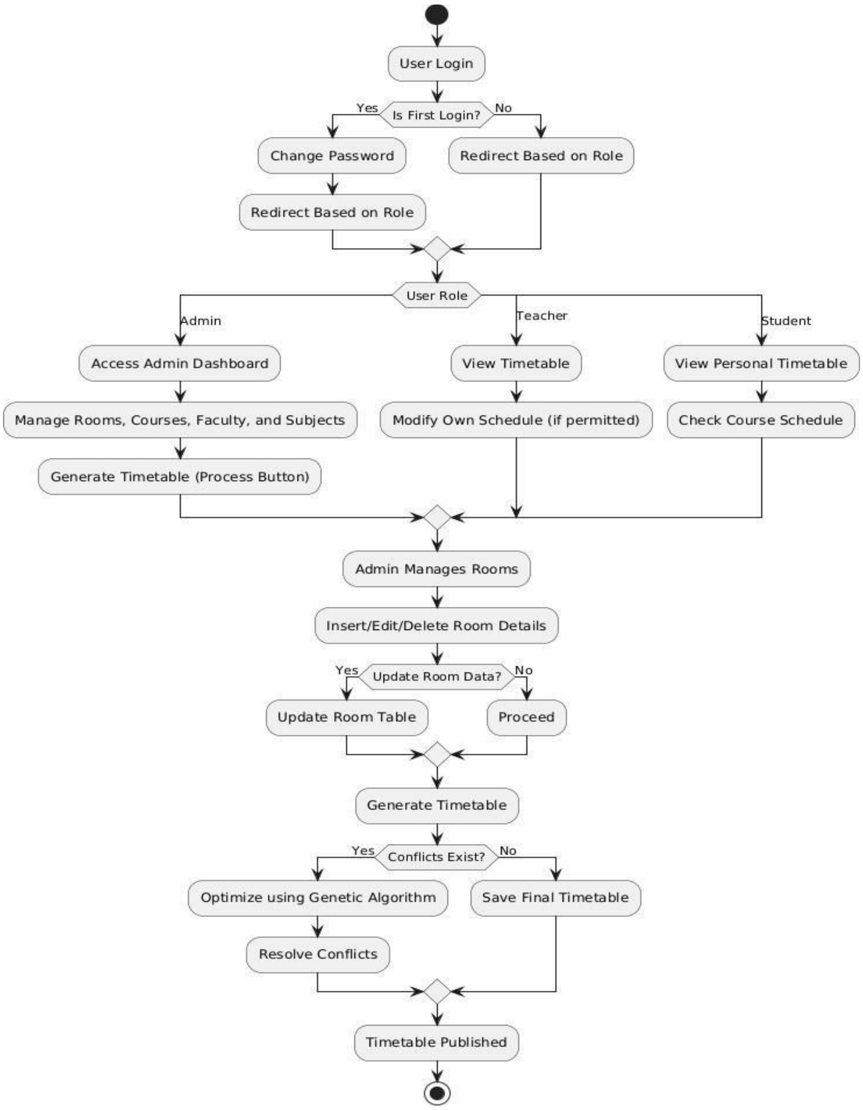


Figure 3.2: System Flowchart

The diagram is a Timetable Management System workflow. It starts with user login, where new users are required to reset their password first. Users are then routed according to their roles: Admin, Teacher, or Student. Admins are able to view the dashboard to manage rooms, courses, faculty, and subjects, as well as produce timetables. Teachers can view their timetables and adjust their schedules if allowed, while students can view their personal timetable and course schedules. The admin handles rooms by adding, modifying, or removing room information. If there is an update of room data, the room table is updated based on it otherwise, timetable generation is continued. If conflicts are detected in generated timetables, they are resolved through a genetic algorithm before it gets finalized. If there are no conflicts, the timetable is stored directly. After verification and resolution, the final timetable is published to users. This systematic method allows for a well-organized and peaceful scheduling system.

3.2 DESIGN PHASE

This phase focuses on converting requirements into a system design that ensures usability, scalability, and efficient data management.

3.2.1 FRONTEND DESIGN

Frontend is designed using JSP, CSS, and JavaScript to ensure an intuitive user interface that supports dynamic content rendering [6].

3.2.2 BACKEND DESIGN

Business logic and server-side processing are structured in Java for robustness. JSP helps with dynamic data interaction. The backend handles scheduling constraints and API communication [1].

3.2.3 DATABASE DESIGN

A normalized MySQL database schema is designed including entities like users, schedules, courses, and feedback. Relationships ensure integrity and prevent scheduling conflicts [1][6][7].

3.2.4 CLASS DIAGRAM

The Smart Timetable System class diagram provided depicts the relationships between various entities involved in timetabling generation and management. User class, with fields like userId, password, role, and isFirstLogin, possesses methods such as login() and changePassword(), and is also categorized into Admin, Student, and Teacher. The Admin can control different elements, such as Rooms, Courses, Faculty, Subjects, and Practicals, through techniques such as manageRooms(), manageCourses(), and manageFaculty(). The Admin also creates the timetable via generateTimetable(). Students and Teachers can viewTimetable(), enabling them to view and verify their timetables. The Timetable class contains properties like timetableId and schedule, and has operations such as generate(), save(), print(), and publish(). The timetable has to be created by the Student, but may be viewed both by Teachers and Students. Different entities like Room, which includes fields like roomId, name, type, capacity, and department, and Course, Faculty, Subject, and Practical classes storing respective information that is needed in scheduling, are handled by Admin. The system structure and system interactions are depicted well by this diagram, giving a proper orderly process of creating and managing a timetable.

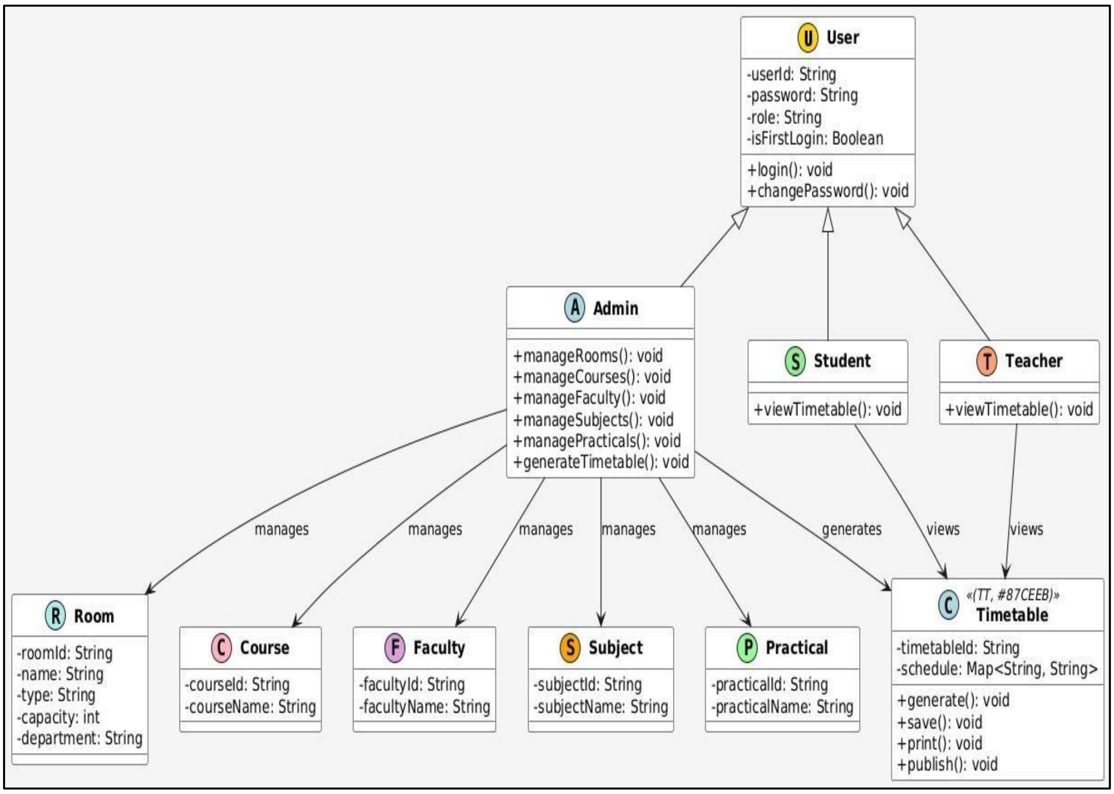


Figure 3.2.4: Class Diagram

3.2.5 COMPONENT DIAGRAM

The component diagram of the Automated Timetable Generation System outlines a comprehensive and modular architecture divided into three primary layers: Frontend, Backend, and Database (MySQL). The Frontend layer consists of four main components: Student View, Teacher View, Admin Dashboard, and Login Interface. These interfaces provide role-specific access and interaction with the system—students and teachers can view their timetables, administrators can manage scheduling operations, and all users authenticate via the login interface. The Backend acts as the core processing unit and houses several interconnected modules. The Timetable Controller is the central orchestrator, receiving requests from the frontend and coordinating with the Genetic Algorithm Engine, which applies evolutionary computing techniques to generate optimal schedules based on constraints. The generated schedules are further verified by the Schedule Conflict Resolver, ensuring there are no clashes in timings, rooms, or faculty availability. The Room Management Module handles the allocation and availability of physical classrooms, while the Authentication Service manages user credentials and secure access. These backend components communicate with the MySQL Database, which stores essential data across five tables: Schedule Table, Course Table, Room Table, User Table, and Feedback Table. The system design promotes scalability, modular development, and efficient data flow, enabling automated, conflict-free, and user-specific timetable generation in educational institutions.

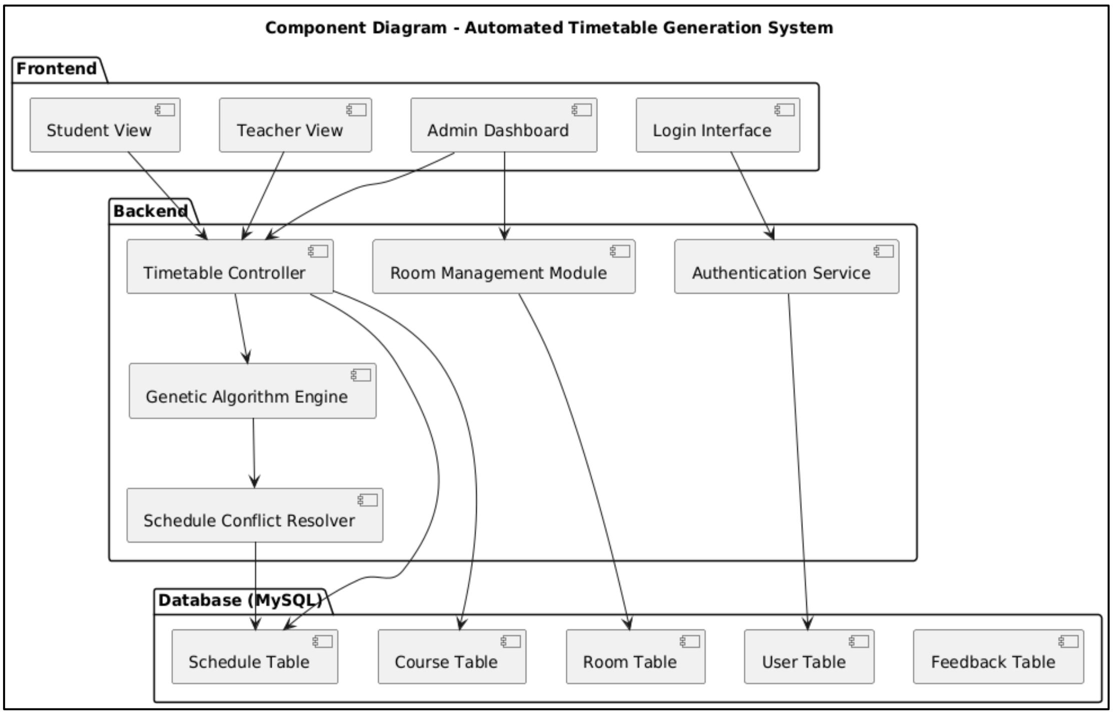


Figure 3.2.5: Component Diagram

3.3 IMPLEMENTATION

In this phase, actual development of frontend, backend, and database integration is carried out. The genetic algorithm is also implemented here.

3.3.1 Frontend Development

Static and dynamic web pages are developed with HTML, CSS, JavaScript, and JSP for seamless user interaction [8].

3.3.2 Backend Development

Java handles business logic, including RESTful API endpoints and genetic algorithm logic for timetable generation [10].

3.3.3 Database Development

MySQL database is implemented to store and manage persistent data like user info, schedules, and feedback [6].

3.3.4 Timetable Generation

A Genetic Algorithm is used to create optimized, conflict-free timetables. Selection, crossover, and mutation operations are applied iteratively [4][6].

3.3.5 BLOCK DIAGRAM

The Smart Timetable Generator comprises several interconnected modules to streamline scheduling. The Dashboard Module offers role-based access via AdminDashboard.jsp, TeacherDashboard.jsp, and StudentDashboard.jsp. The Management Module handles academic data such as rooms, courses, faculty, and subjects through dedicated JSP pages. Secure login and password changes are managed by the Authentication Module using Login.jsp, LoginServlet, and ChangePasswordServlet. The Timetable Generation Module automates scheduling through SetupTimetable.jsp, GenerateProcess.jsp, ProcessTimetable.jsp, and ShowTimetable.jsp. Post-generation, the Timetable Management Module allows admins to edit, save, print, and publish schedules via servlets like EditServlet, SaveServlet, PrintServlet, and PublishServlet. All essential data is stored in the Database, and the system follows a structured flow—from authentication and data retrieval to generation and publishing—for efficient and organized timetable management.

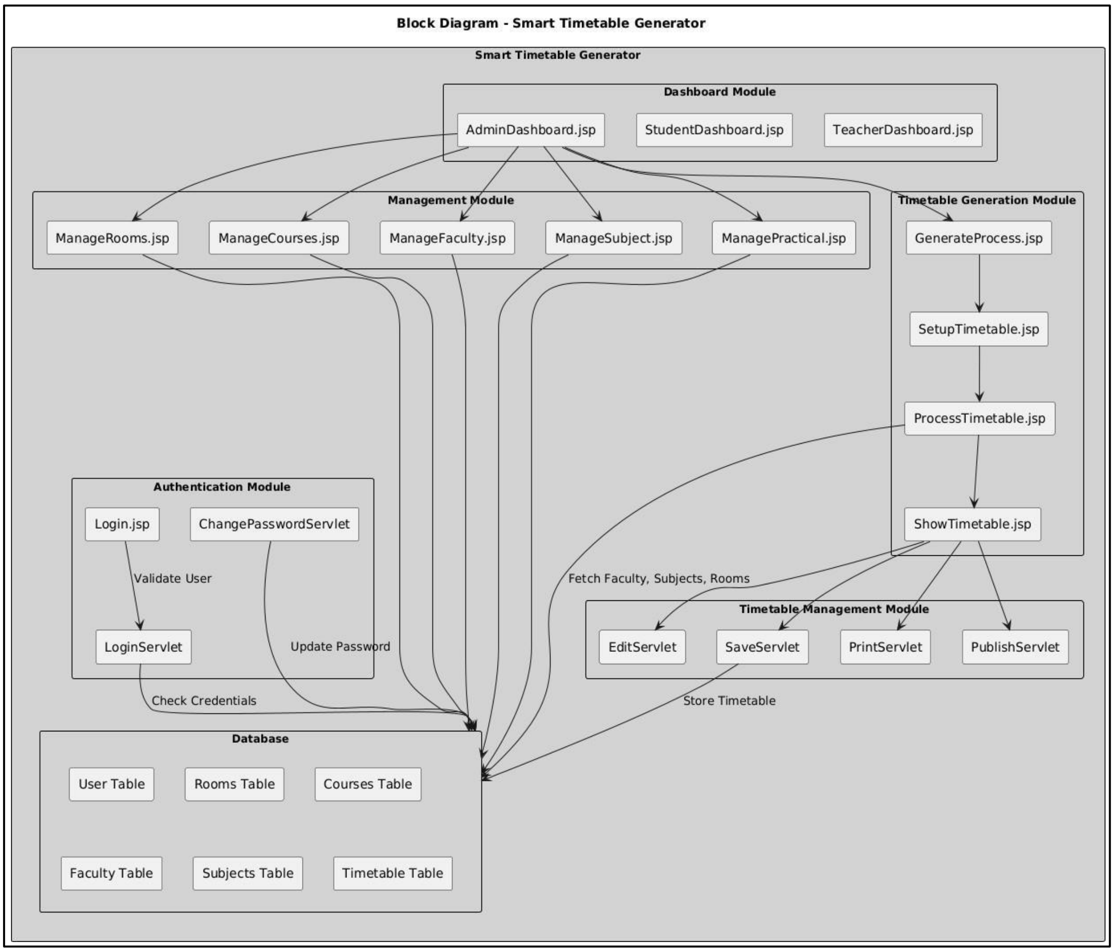


Figure 3.3.5: Block Diagram

3.3.6 GENETIC ALGORITHM:

Five phases are typically involved in a genetic algorithm.

1. Initialization: Generate an initial population of chromosomes.
2. Evaluation: Evaluate the fitness of each chromosome in the population using the fitness function.
3. Selection: Select chromosomes from the current population for reproduction based on their fitness.
4. Crossover: Generate offspring by combining genetic material from selected parent chromosomes.
5. Mutation: Introduce random changes to the offspring chromosomes to maintain diversity.

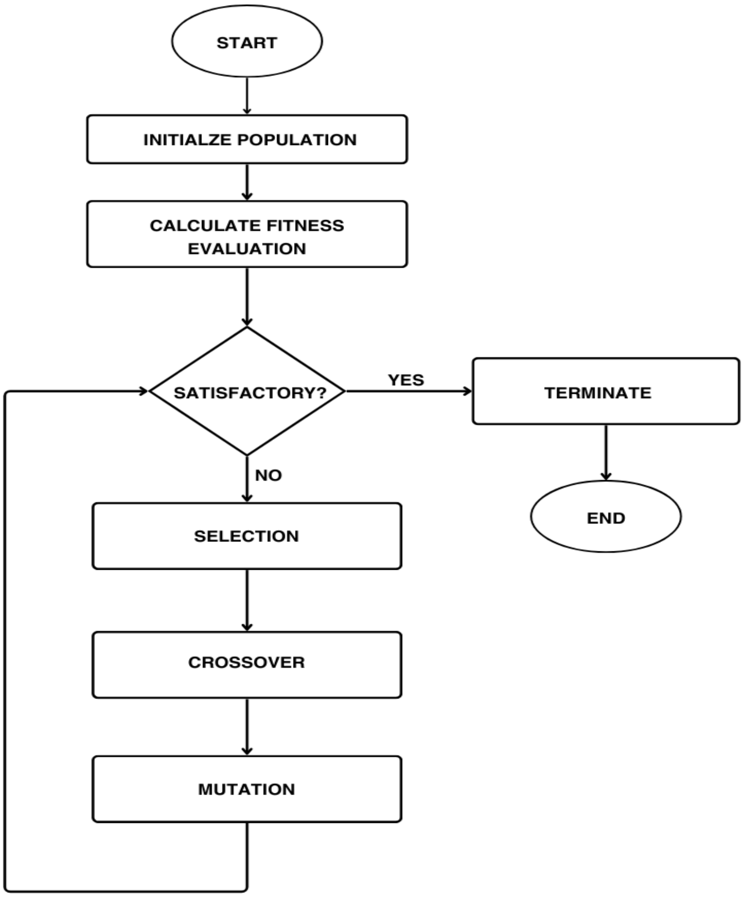


Figure 3.3.6: Genetic Algorithm

3.4 TESTING

Testing ensures system reliability, correctness, and performance before deployment.

3.4.1 UNIT TESTING

Individual units (functions, API methods, modules) are tested for correctness [5].

3.4.2 INTEGRATION TESTING

Interaction between components (UI, backend, database) is validated to ensure smooth data flow [1].

3.4.3 SYSTEM TESTING

Overall system is tested for performance and usability under real-world conditions [5].

3.4.4 USER ACCEPTANCE TESTING

End-users (admin, teacher, student) test the system for usability, and feedback is gathered for improvements [3].

3.4.5 SEQUENCE DIAGRAM

Smart Timetable Generator - Sequence Diagram illustrates the coordination among a user, multiple system components, and the database across various processes. It has multiple phases beginning with the User Login Process, where the user inputs credentials through Login.jsp, which are then passed on to LoginServlet for authentication. After verification, the system identifies the user role and directs the user to the respective dashboard. In the Managing Rooms (Admin) phase, the admin uses the ManageRooms.jsp page, which sends a request and receives room information from the database by using RoomServlet. The admin can add, update, or remove room information, and any changes are updated in the database prior to being presented to the user. Then, during the Generating Timetable phase, the user visits GenerateTimetable.jsp, which initiates timetable generation. Process TimetableServlet fetches faculty, subject, and room information from the database, processes it, and generates the timetable for display. During the Saving Timetable phase, the saved timetable is stored through SaveTimetable.jsp, which saves the timetable information.

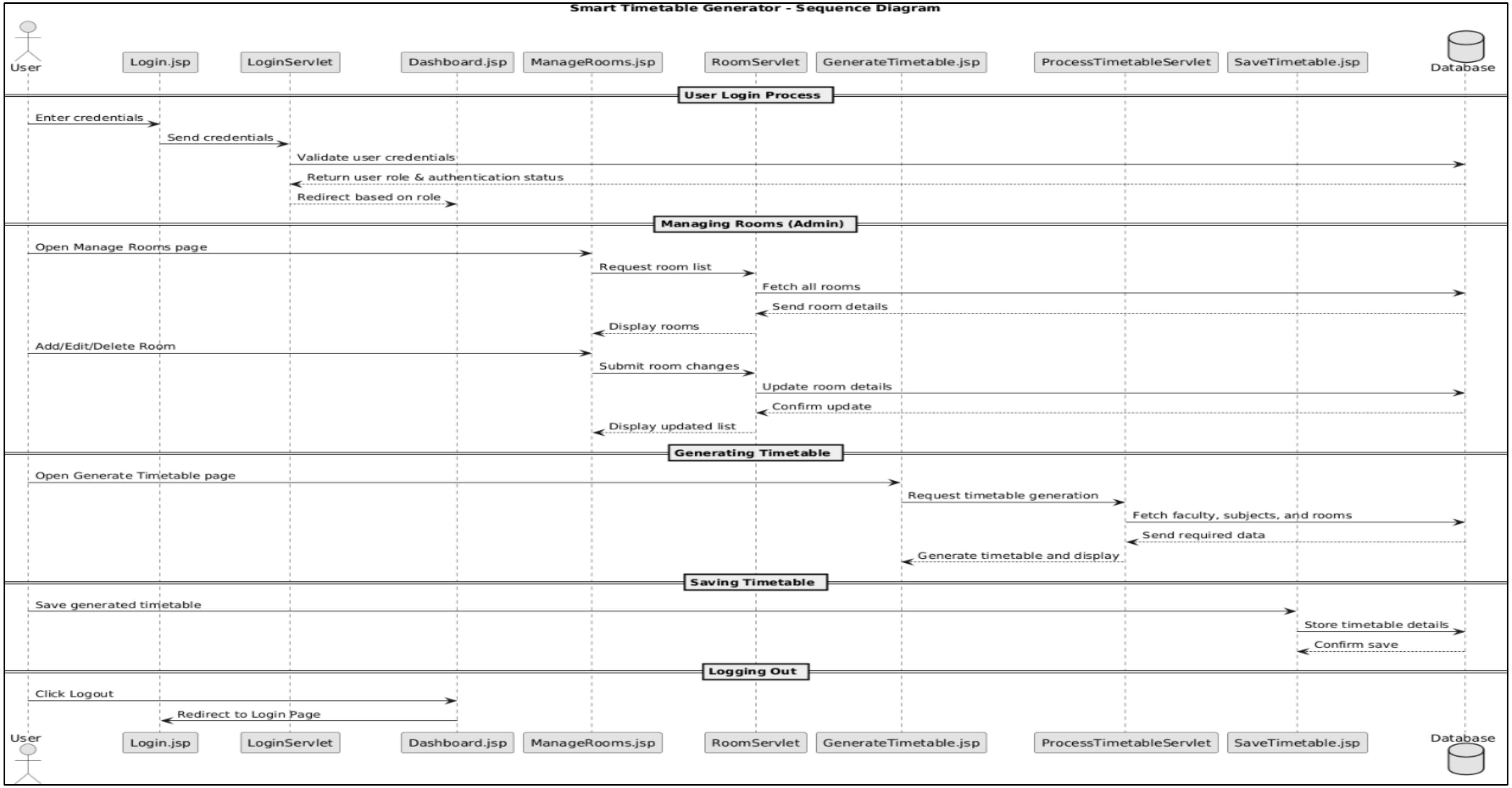


Figure 3.4.5: Sequence Diagram

3.5 DEPLOYMENT

After successful testing, the system is deployed in a production environment.

3.5.1 PREPARE DEPLOYMENT ENVIRONMENT

Setup OF Java, MySQL, Apache Tomcat, and security configurations (e.g., SSL) [6].

3.5.2 APPLICATION DEPLOYMENT

Frontend, backend, and database components are deployed and made accessible via secure URL [2].

3.5.3 FINAL TESTING

System undergoes final performance and security testing to ensure stability in the production environment [5][6].

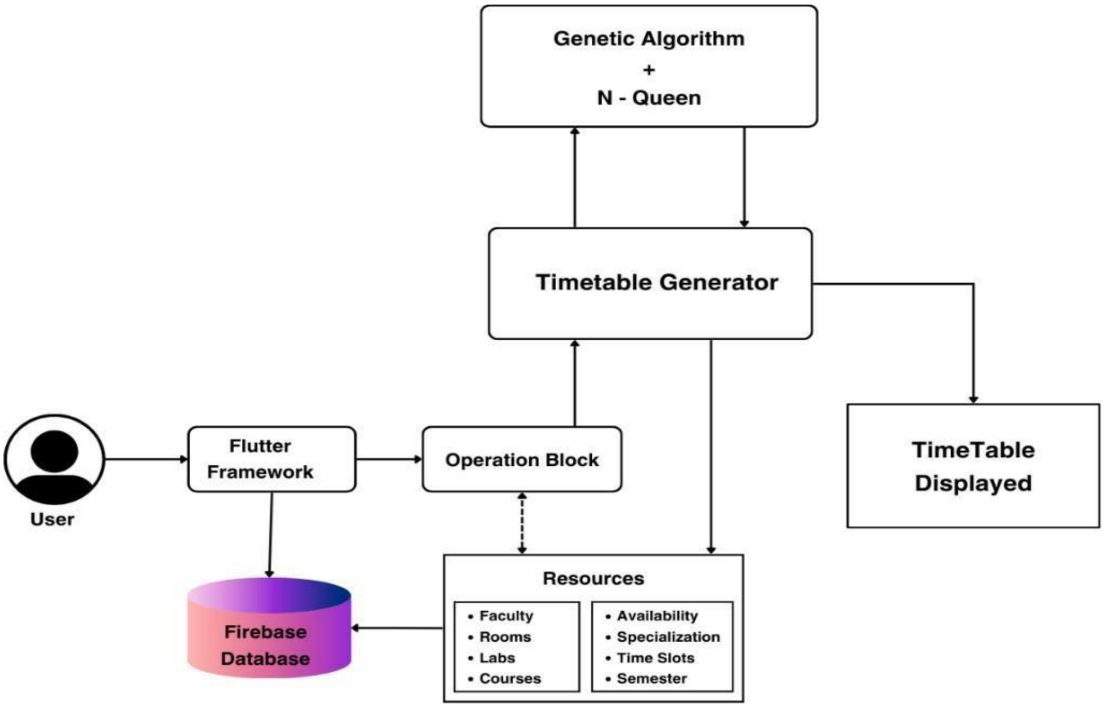


Figure 3.5.1: Existing System Architecture

The system architecture diagram given above illustrates the process of Smart Timetable Generation employing Genetic Algorithm and N-Queen Problem integrated together for effective scheduling. The system is comprised of various interlinked components that enable automated generation of timetables. The user is given a Flutter-based front-end app with which to interact with the system, sending necessary scheduling materiallike faculty schedules, rooms, labs, courses, specializations, time periods, and semesters to a Firebase database, where it retrieves these resources to be sent on to the Operation Block. It is here in the Operation Block that the information is processed, ensuring seamless integration between the front-end and timetable generation code. The fundamental scheduling algorithm, based on Genetic Algorithm and N-Queen, maximizes timetable generation by minimizing conflicts and allocating slots efficiently. The Timetable Generator then executes these optimizations and produces a formatted schedule. Lastly, the produced timetable is presented to the user, resulting in an organized and conflict-free scheduling system. The modularity and automation of this architecture enhance efficiency, minimize manual effort, and increase flexibility for institutional requirements.

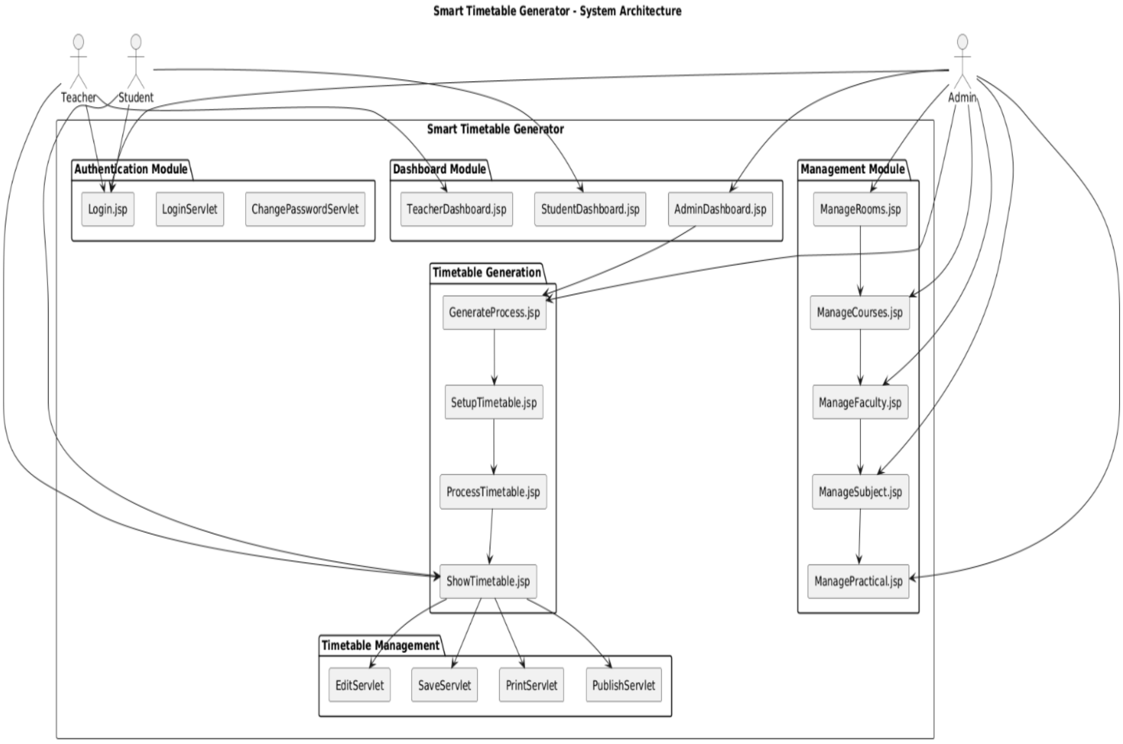


Figure 3.5.2: Proposed System Architecture

The system architecture of the proposed Smart Timetable Generator is aimed at simplifying the process of creating and managing timetables for schools. It is made up of several modules, each with its own set of functionalities. The Authentication Module provides safe user access through login and password modification management. The Dashboard Module offers role-based access, where teachers, students, and administrators can access their respective interfaces. The Management Module facilitates administrators to control critical resources like rooms, courses, faculty, subjects, and practical sessions.

CHAPTER 4

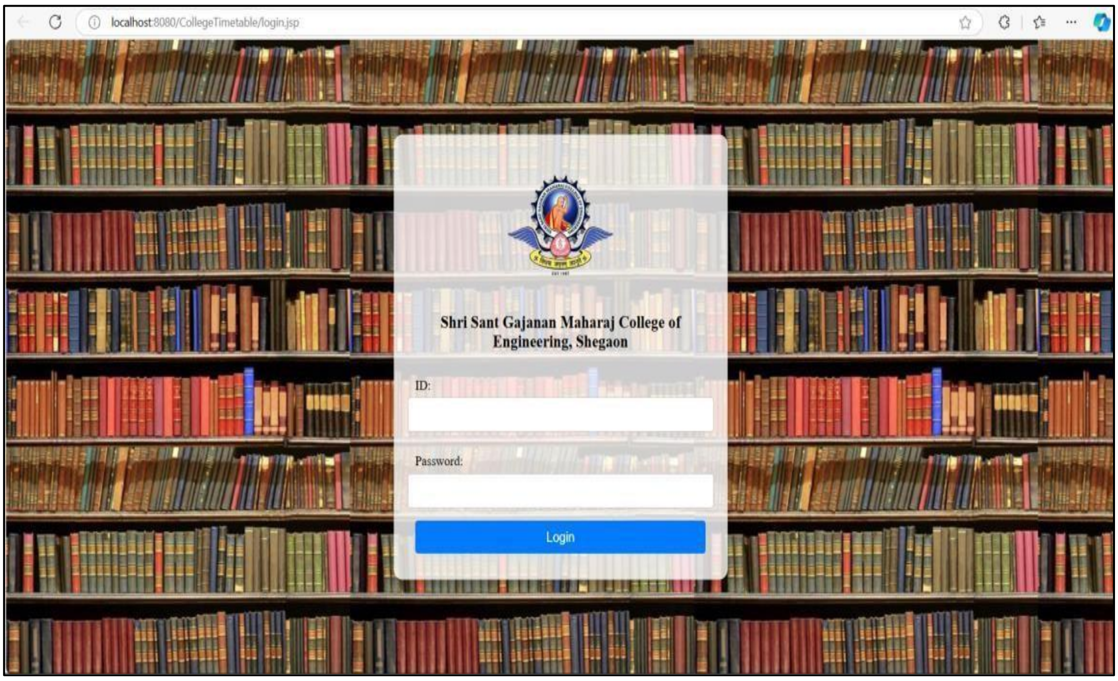
IMPLEMENTATION

IMPLEMENTATION

The implementation of the Automated Timetable Generation System is done using Java Servlet for handling backend logic such as request processing, user authentication, and timetable generation. JSP (JavaServer Pages) is used for rendering dynamic content and creating an interactive user interface for admins, teachers, and students. MySQL serves as the relational database, storing all essential data like users, rooms, courses, and schedules. The backend logic also includes a genetic algorithm implemented in Java to generate optimized, conflict-free timetables. These components work together to deliver a responsive and efficient academic scheduling system.

4.1 USER MANAGEMENT

User administration in this system provides safe and role-based access to the college timetable portal. The login screen prompts users to provide their credentials (ID and Password), following which the system verifies the provided input and leads them to their respective dashboards depending on their roles—Teacher, Student, or Admin. Teachers can control class timetables and see assigned lectures, Students can see their own individual timetables, and Admins have complete control over user management, timetables, and system settings. This organization provides streamlined, structured access and operations that are specific to the requirements of each user group.


localhost:8080/CollegeTimetable/login.jsp
Shri Sant Gajanan Maharaj College of Engineering, Shegaon
ID:
Password:
Login

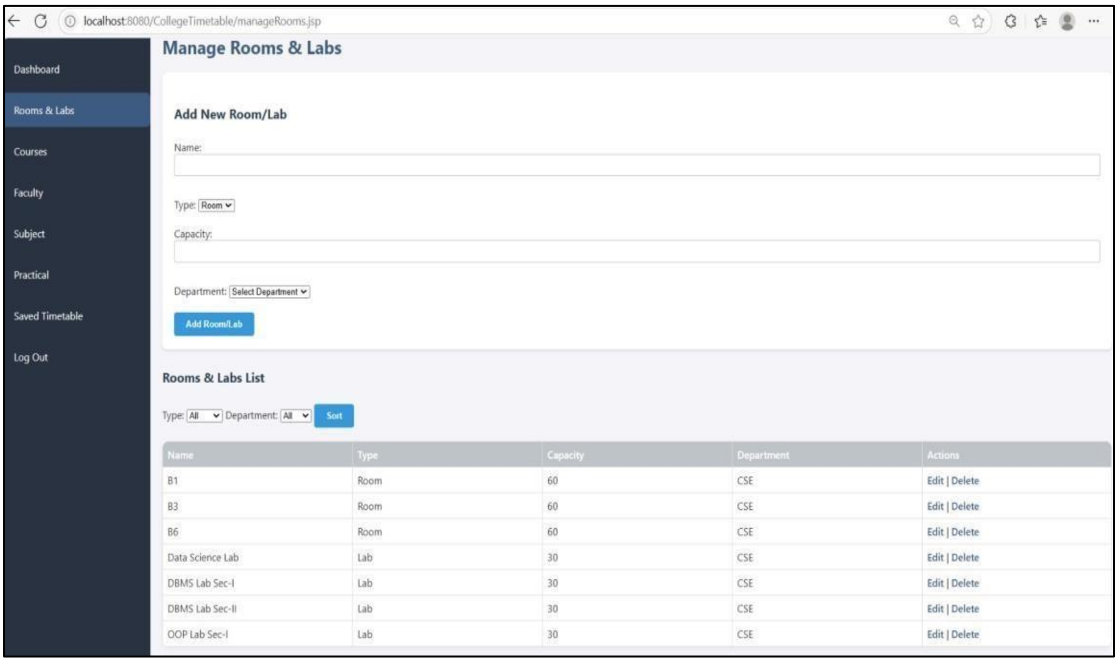
Screenshot 4.1: Login Page

4.2 DATA INPUT

The Data Input module gathers vital information needed for efficient timetable generation. This includes detailed information regarding rooms and labs (like room numbers, capacities, and types), courses taught during the academic term, subjects of each course, practical sessions with batch-wise allocations, and faculty information including their subject expertise and availability. This organized data provides the core for the scheduling mechanism, guaranteeing that the timetable produced matches the academic demands and resource availability requirements while also accommodating user-defined options.

4.2.1 ROOMS & LABS

Data can be entered by users about available labs and rooms, including their names, sizes, and types (e.g., theory or for practice). Every entry can be modified to update changes or removed if no longer needed.



Manage Rooms & Labs

Add New Room/Lab

Name:

Type: Room

Capacity:

Department: Select Department

Rooms & Labs List

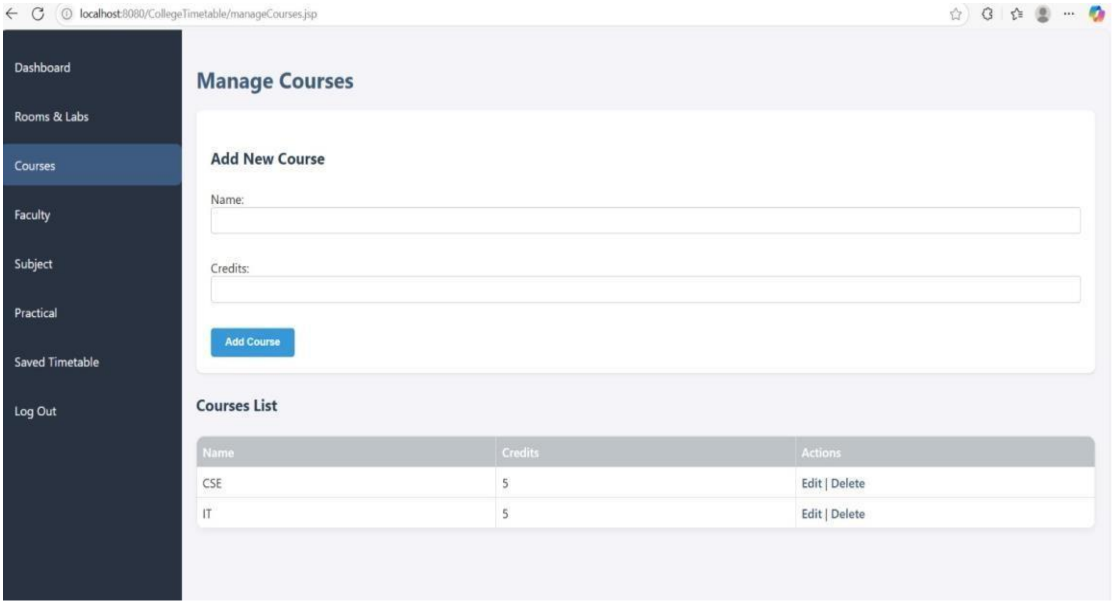
Type: All Department: All

Name	Type	Capacity	Department	Actions
B1	Room	60	CSE	Edit | Delete
B3	Room	60	CSE	Edit | Delete
B6	Room	60	CSE	Edit | Delete
Data Science Lab	Lab	30	CSE	Edit | Delete
DBMS Lab Sec-I	Lab	30	CSE	Edit | Delete
DBMS Lab Sec-II	Lab	30	CSE	Edit | Delete
OOP Lab Sec-I	Lab	30	CSE	Edit | Delete

Screenshot 4.2: Manage Rooms & Labs

4.2.2 COURSES

Users can add diverse academic programs (e.g., B.E., M.E.) taught in the institution. Course names can be updated or old ones can be removed as necessary.



Manage Courses

Add New Course

Name:

Credits:

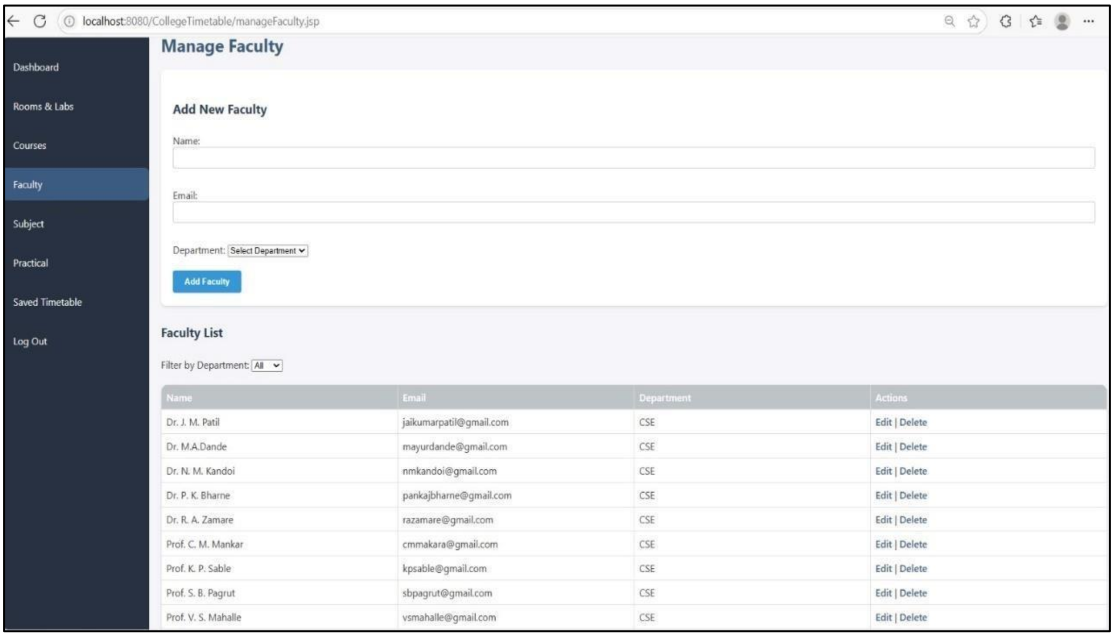
Courses List

Name	Credits	Actions
CSE	5	Edit | Delete
IT	5	Edit | Delete

Screenshot 4.3: Manage Courses

4.2.3 FACULTY

Details of faculty including names, allotted subjects, departments, and slots available are inputted here. Admins can edit this information to provide updates on changed faculty or delete records when a faculty member is inactive.



Manage Faculty

Add New Faculty

Name:

Email:

Department: Select Department

Faculty List

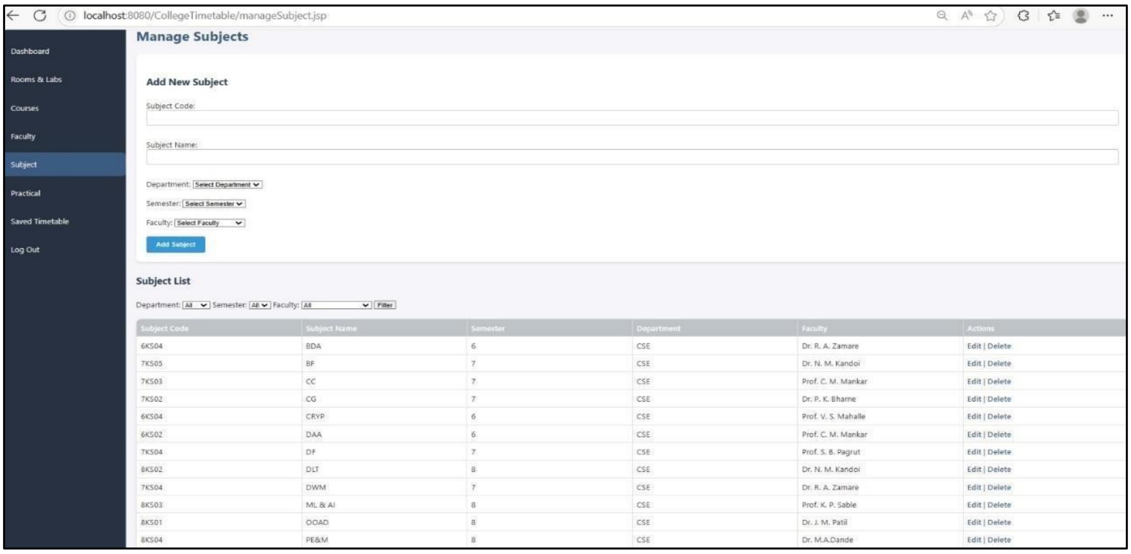
Filter by Department: All

Name	Email	Department	Actions
Dr. J. M. Patil	jaikumarpatil@gmail.com	CSE	Edit | Delete
Dr. M.A.Dande	mayurdande@gmail.com	CSE	Edit | Delete
Dr. N. M. Kandoi	nmkandoi@gmail.com	CSE	Edit | Delete
Dr. P. K. Bharne	pankajbharne@gmail.com	CSE	Edit | Delete
Dr. R. A. Zamare	razamare@gmail.com	CSE	Edit | Delete
Prof. C. M. Mankar	cmmakara@gmail.com	CSE	Edit | Delete
Prof. K. P. Sable	kpsable@gmail.com	CSE	Edit | Delete
Prof. S. B. Pagrut	sbpagrut@gmail.com	CSE	Edit | Delete
Prof. V. S. Mahalle	vsmahalle@gmail.com	CSE	Edit | Delete

Screenshot 4.4: Manage Faculty

4.2.4 SUBJECTS

Subject under each course and semester can be included, such as theory and elective subjects. Editing features enable updates in subject names or codes, and deletion to remove discontinued subjects.



localhost:8080/CollegeTimetable/manageSubject.jsp

Manage Subjects

Add New Subject

Subject Code:

Subject Name:

Department: Select Department

Semester: Select Semester

Faculty: Select Faculty

Subject List

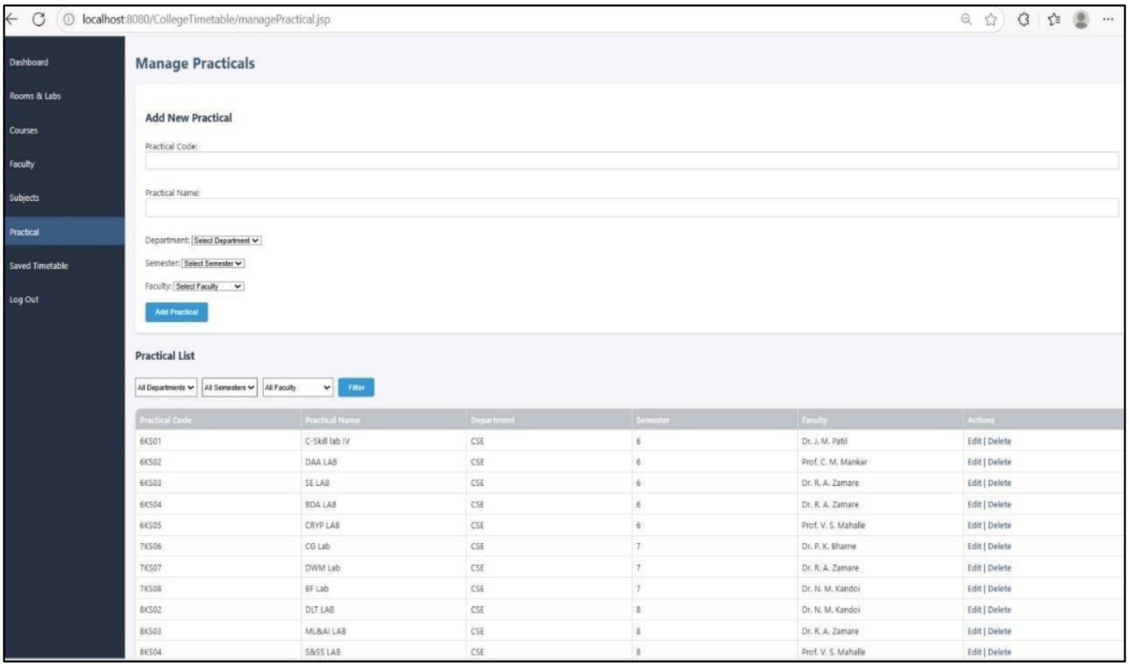
Department: All Semester: All Faculty: All

Subject Code	Subject Name	Semester	Department	Faculty	Actions
6KS04	BDA	6	CSE	Dr. R. A. Zamare	Edit | Delete
7KS05	BF	7	CSE	Dr. N. M. Kandoi	Edit | Delete
7KS03	CC	7	CSE	Prof. C. M. Mankar	Edit | Delete
7KS02	CG	7	CSE	Dr. P. K. Bharne	Edit | Delete
6KS04	CRYP	6	CSE	Prof. V. S. Mahalle	Edit | Delete
6KS02	DAA	6	CSE	Prof. C. M. Mankar	Edit | Delete
7KS04	DF	7	CSE	Prof. S. B. Pagrut	Edit | Delete
8KS02	DLT	8	CSE	Dr. N. M. Kandoi	Edit | Delete
7KS04	DWM	7	CSE	Dr. R. A. Zamare	Edit | Delete
8KS03	ML & AI	8	CSE	Prof. K. P. Sable	Edit | Delete
8KS01	OOAD	8	CSE	Dr. J. M. Patil	Edit | Delete
8KS04	PE&M	8	CSE	Dr. M.A.Dande	Edit | Delete

Screenshot 4.5: Manage Subjects

4.2.5 PRACTICALS

Practical sessions can be created with details such as subject name, batch divisions, and lab requirements. These entries can be updated or deleted to support timetable changes.



localhost:8080/CollegeTimetable/managePractical.jsp

Manage Practicals

Add New Practical

Practical Code:

Practical Name:

Department: Select Department

Semester: Select Semester

Faculty: Select Faculty

Practical List

All Departments All Semesters All Faculty

Practical Code	Practical Name	Department	Semester	Faculty	Actions
6KS01	C-Skill lab IV	CSE	6	Dr. J. M. Patil	Edit | Delete
6KS02	DAA LAB	CSE	6	Prof. C. M. Mankar	Edit | Delete
6KS03	SE LAB	CSE	6	Dr. R. A. Zamare	Edit | Delete
6KS04	BDA LAB	CSE	6	Dr. R. A. Zamare	Edit | Delete
6KS05	CRYP LAB	CSE	6	Prof. V. S. Mahalle	Edit | Delete
7KS06	CG Lab	CSE	7	Dr. P. K. Bharne	Edit | Delete
7KS07	DWM Lab	CSE	7	Dr. R. A. Zamare	Edit | Delete
7KS08	BF Lab	CSE	7	Dr. N. M. Kandoi	Edit | Delete
8KS02	DLT LAB	CSE	8	Dr. N. M. Kandoi	Edit | Delete
8KS03	ML&AI LAB	CSE	8	Dr. R. A. Zamare	Edit | Delete
8KS04	S&SS LAB	CSE	8	Prof. V. S. Mahalle	Edit | Delete

Screenshot 4.6: Manage Practicals

4.3 TIMETABLE GENERATION

Timetable Generation is the central functionality of our system, where class timetables are generated using Genetic Algorithms (GA) to maximize resource utilization while meeting academic and logistical requirements.

4.3.1 GENETIC ALGORITHMIC APPROACH

The generation of the timetable utilizes a Genetic Algorithm, an evolutionary computation method based on natural selection. It starts with a random initial population of timetable solutions and repeatedly improves them through processes such as selection, crossover, and mutation. Each timetable is assessed according to a fitness function that takes into account faculty availability, classroom capacities, and batch student preference constraints. Across generations, the algorithm tends to converge to very optimal and viable timetables.

4.3.2 RESOURCE ALLOCATION

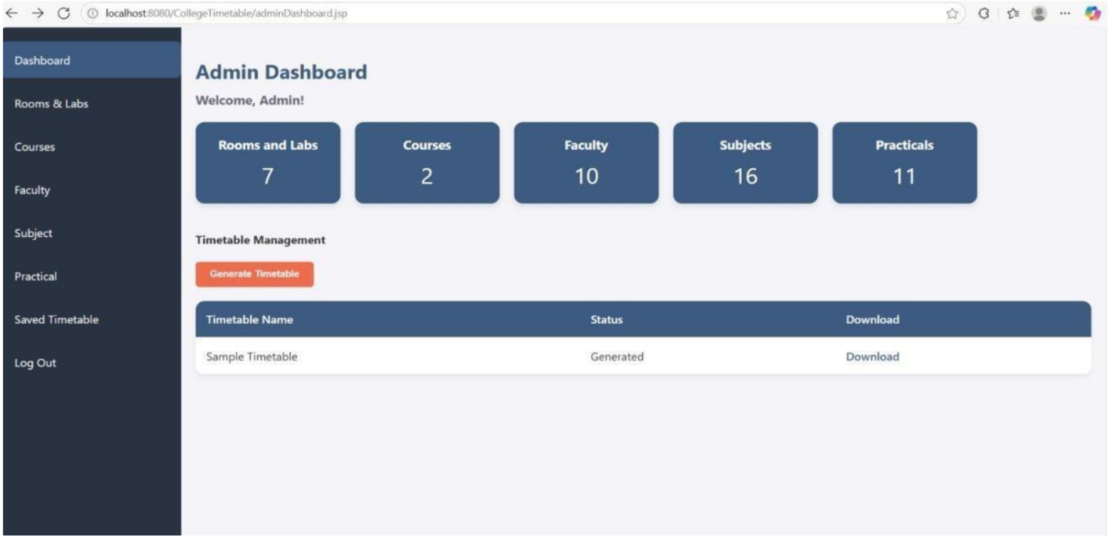
The Genetic Algorithm optimally allocates resources like faculty members, classrooms, and lab time slots to subjects and practicals according to their availability and institutional limitations. The algorithm focuses on conflict-free, efficient scheduling to minimize faculty idle time and reduce room clashes to ensure optimal utilization of available resources.

4.3.3 CONFLICT RESOLUTION

Conflicts like double-booked rooms, double-booked lectures, or no faculty slots available are penalized as conflicts in the fitness function. The Genetic Algorithm goes around or minimizes these penalties by learning improved solutions with each generation until it yields valid, conflict-free timetables that honor all soft and hard constraints.

4.4 USER INTERFACE

The User Interface provides a user-friendly platform for interacting with the system, facilitating data input and timetable viewing.



Admin Dashboard

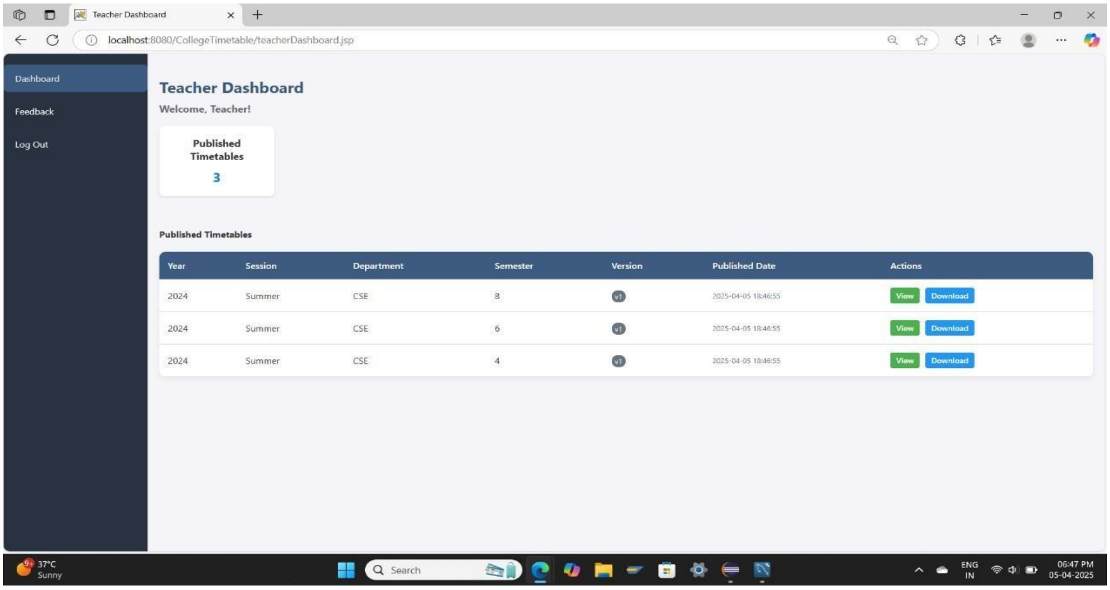
Welcome, Admin!

Rooms and Labs	Courses	Faculty	Subjects	Practicals
7	2	10	16	11

Timetable Management

Timetable Name	Status	Download
Sample Timetable	Generated	Download

Screenshot 4.7: Admin Dashboard



Teacher Dashboard

Welcome, Teacher!

Published Timetables
3

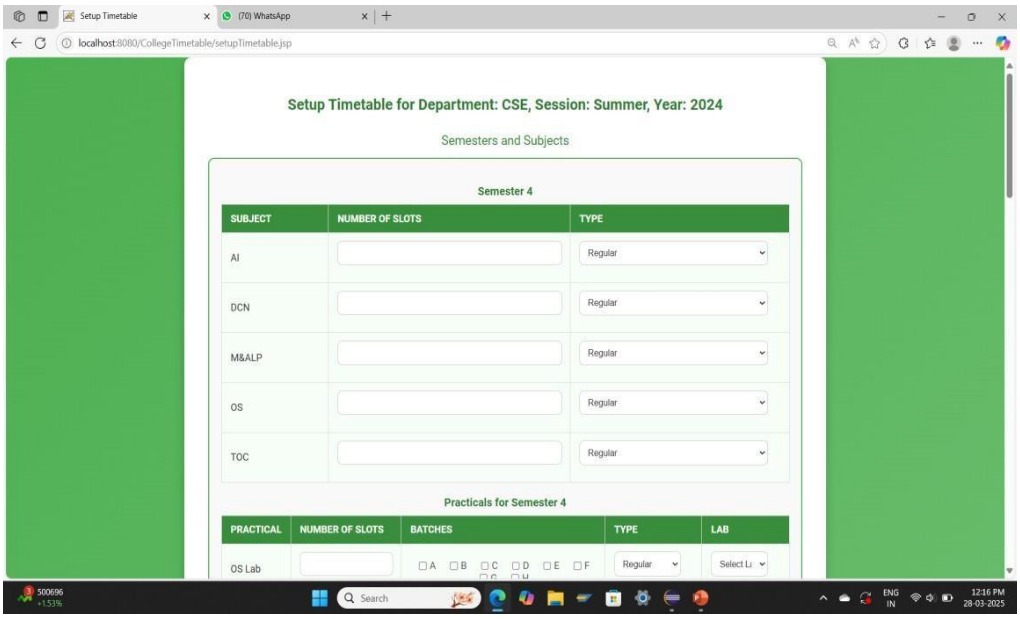
Published Timetables

Year	Session	Department	Semester	Version	Published Date	Actions
2024	Summer	CSE	8	v1	2025-04-05 18:46:55	View Download
2024	Summer	CSE	6	v1	2025-04-05 18:46:55	View Download
2024	Summer	CSE	4	v1	2025-04-05 18:46:55	View Download

Screenshot 4.8: Teacher Dashboard

4.4.1 INPUT FORMS

Intuitive input forms allow users to input their preferences and requirements easily. Clear instructions and user-friendly controls enhance the usability of the interface.



Setup Timetable for Department: CSE, Session: Summer, Year: 2024

Semesters and Subjects

Semester 4

SUBJECT	NUMBER OF SLOTS	TYPE
AI		Regular
DCN		Regular
M&ALP		Regular
OS		Regular
TOC		Regular

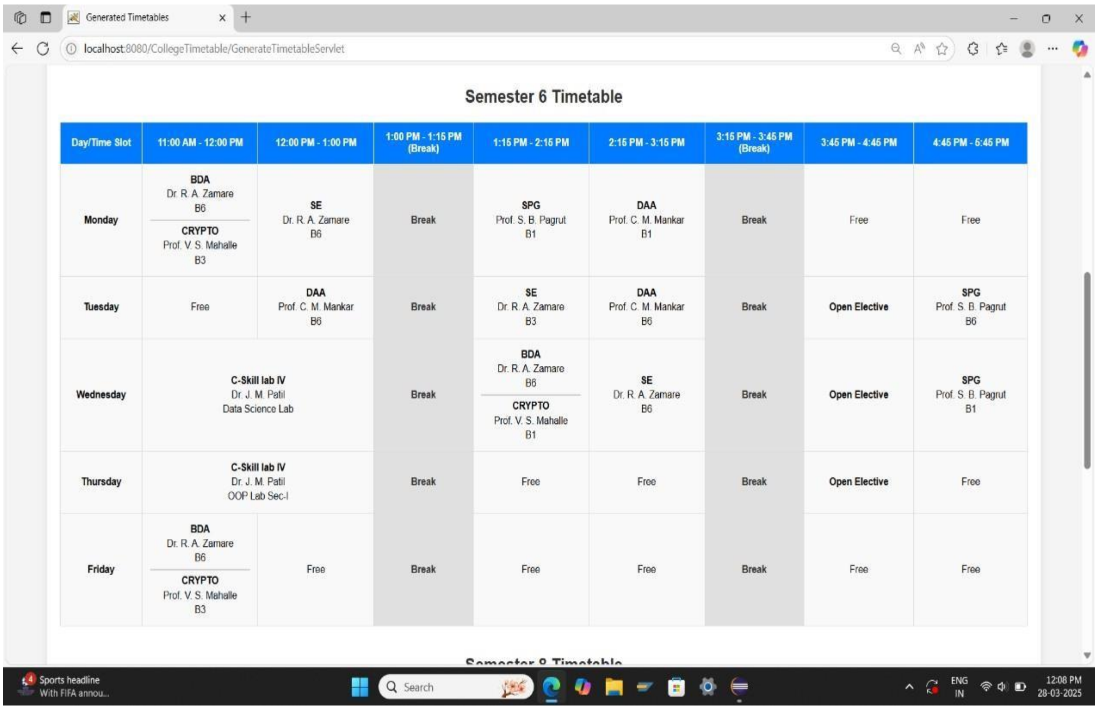
Practicals for Semester 4

PRACTICAL	NUMBER OF SLOTS	BATCHES	TYPE	LAB
OS Lab		□ A □ B □ C □ D □ E □ F □ G □ H	Regular	Select La

Screenshot 4.9: Information Fill Up Page

4.4.2 TIMETABLE DISPLAY

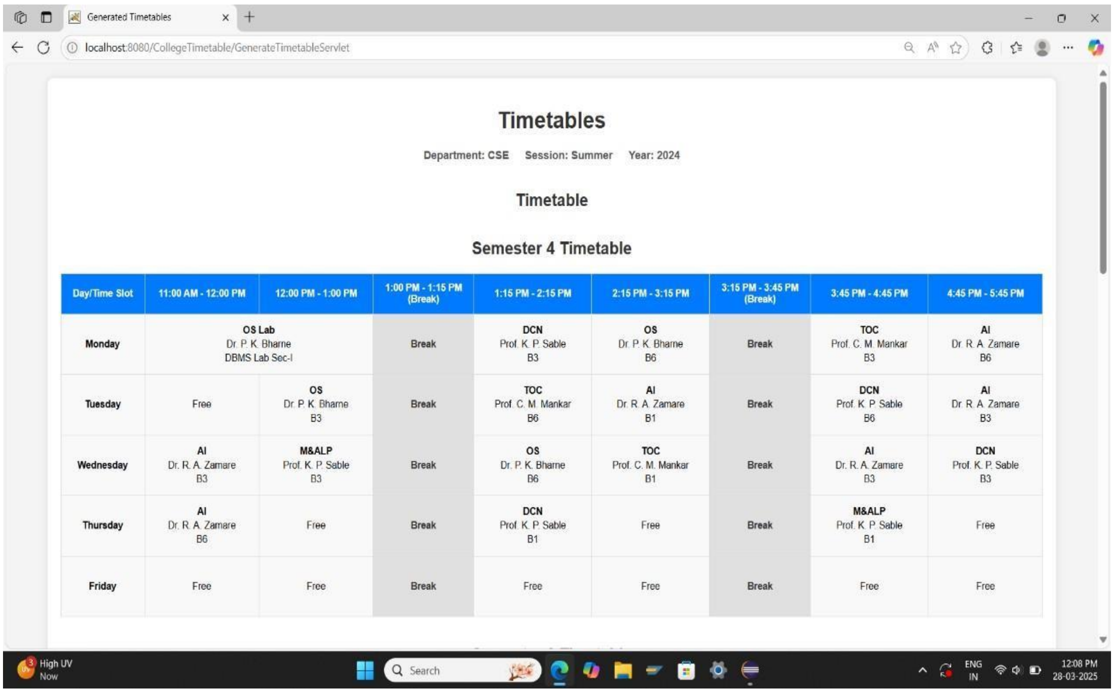
Generated timetables are displayed in a visually appealing and easy-to-understand format. Color-coded schedules, interactive features, and filtering options enhance the user experience and make it easy to navigate through the timetable.



Semester 6 Timetable

Day/Time Slot	11:00 AM - 12:00 PM	12:00 PM - 1:00 PM	1:00 PM - 1:15 PM (Break)	1:15 PM - 2:15 PM	2:15 PM - 3:15 PM	3:15 PM - 3:45 PM (Break)	3:45 PM - 4:45 PM	4:45 PM - 5:45 PM
Monday	BDA Dr. R. A. Zamare B6 CRYPTO Prof. V. S. Mahalle B3	SE Dr. R. A. Zamare B6	Break	SPG Prof. S. B. Pagrut B1	DAA Prof. C. M. Mankar B1	Break	Free	Free
Tuesday	Free	DAA Prof. C. M. Mankar B6	Break	SE Dr. R. A. Zamare B3	DAA Prof. C. M. Mankar B6	Break	Open Elective	SPG Prof. S. B. Pagrut B6
Wednesday	C-Skill lab IV Dr. J. M. Patil Data Science Lab		Break	BDA Dr. R. A. Zamare B6 CRYPTO Prof. V. S. Mahalle B1	SE Dr. R. A. Zamare B6	Break	Open Elective	SPG Prof. S. B. Pagrut B1
Thursday	C-Skill lab IV Dr. J. M. Patil OOP Lab Sec-I		Break	Free	Free	Break	Open Elective	Free
Friday	BDA Dr. R. A. Zamare B6 CRYPTO Prof. V. S. Mahalle B3	Free	Break	Free	Free	Break	Free	Free

Semester 8 Timetable



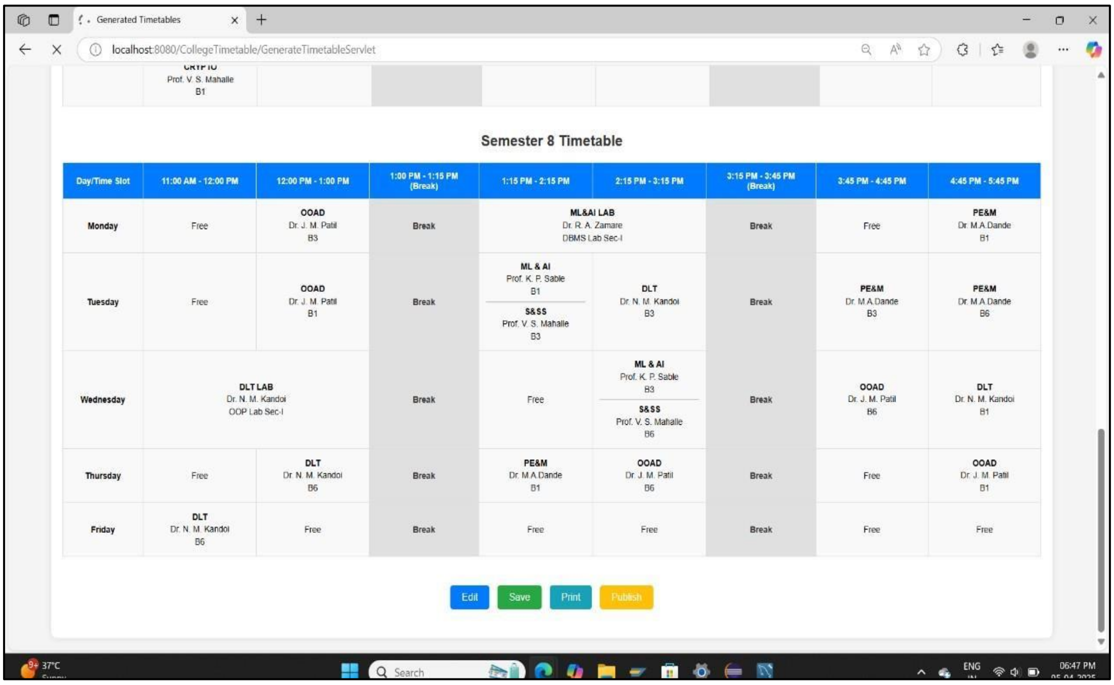
Timetables

Department: CSE Session: Summer Year: 2024

Timetable

Semester 4 Timetable

Day/Time Slot	11:00 AM - 12:00 PM	12:00 PM - 1:00 PM	1:00 PM - 1:15 PM (Break)	1:15 PM - 2:15 PM	2:15 PM - 3:15 PM	3:15 PM - 3:45 PM (Break)	3:45 PM - 4:45 PM	4:45 PM - 5:45 PM
Monday	OS Lab Dr. P. K. Bharne DBMS Lab Sec-I		Break	DCN Prof. K. P. Sable B3	OS Dr. P. K. Bharne B6	Break	TOC Prof. C. M. Mankar B3	AI Dr. R. A. Zamare B6
Tuesday	Free	OS Dr. P. K. Bharne B3	Break	TOC Prof. C. M. Mankar B6	AI Dr. R. A. Zamare B1	Break	DCN Prof. K. P. Sable B6	AI Dr. R. A. Zamare B3
Wednesday	AI Dr. R. A. Zamare B3	M&ALP Prof. K. P. Sable B3	Break	OS Dr. P. K. Bharne B6	TOC Prof. C. M. Mankar B1	Break	AI Dr. R. A. Zamare B3	DCN Prof. K. P. Sable B3
Thursday	AI Dr. R. A. Zamare B6	Free	Break	DCN Prof. K. P. Sable B1	Free	Break	M&ALP Prof. K. P. Sable B1	Free
Friday	Free	Free	Break	Free	Free	Break	Free	Free



CRYPTO
Prof. V. S. Mahalle
B1

Semester 8 Timetable

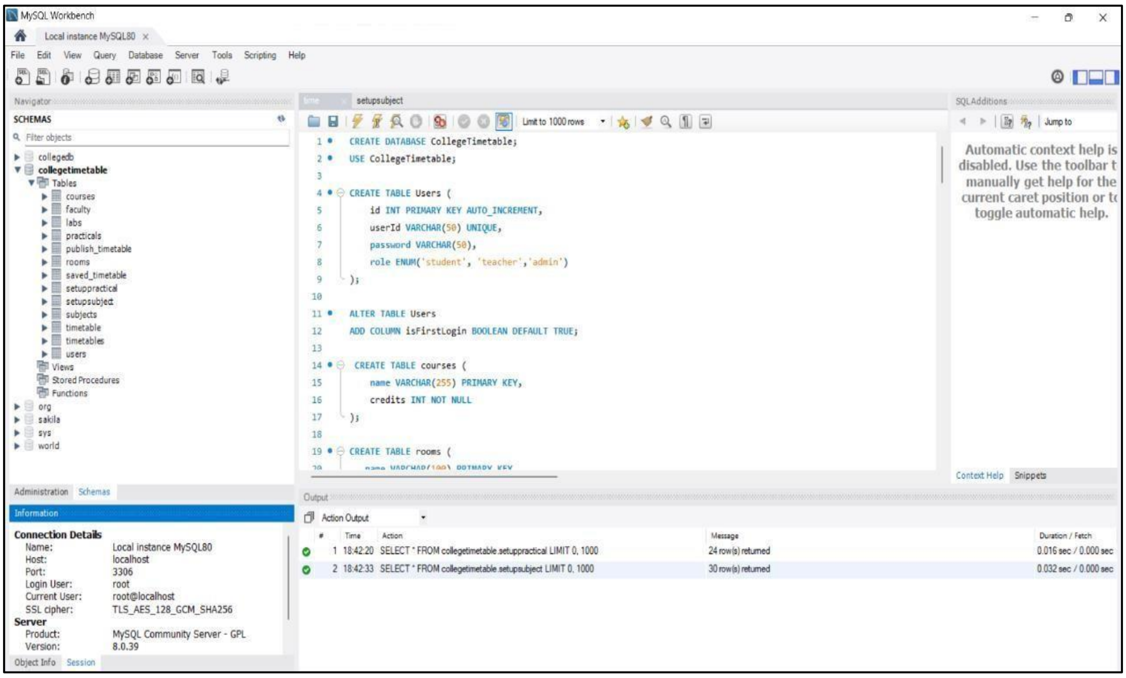
Day/Time Slot	11:00 AM - 12:00 PM	12:00 PM - 1:00 PM	1:00 PM - 1:15 PM (Break)	1:15 PM - 2:15 PM	2:15 PM - 3:15 PM	3:15 PM - 3:45 PM (Break)	3:45 PM - 4:45 PM	4:45 PM - 5:45 PM
Monday	Free	OOAD Dr. J. M. Patil B3	Break	ML&AI LAB Dr. R. A. Zamare DBMS Lab Sec-I		Break	Free	PE&M Dr. M.A.Dande B1
Tuesday	Free	OOAD Dr. J. M. Patil B1	Break	ML & AI Prof. K. P. Sable B1 S&SS Prof. V. S. Mahalle B3	DLT Dr. N. M. Kandoi B3	Break	PE&M Dr. M.A.Dande B3	PE&M Dr. M.A.Dande B6
Wednesday	DLT LAB Dr. N. M. Kandoi OOP Lab Sec-I		Break	Free	ML & AI Prof. K. P. Sable B3 S&SS Prof. V. S. Mahalle B6	Break	OOAD Dr. J. M. Patil B6	DLT Dr. N. M. Kandoi B1
Thursday	Free	DLT Dr. N. M. Kandoi B6	Break	PE&M Dr. M.A.Dande B1	OOAD Dr. J. M. Patil B6	Break	Free	OOAD Dr. J. M. Patil B1
Friday	DLT Dr. N. M. Kandoi B6	Free	Break	Free	Free	Break	Free	Free

Screenshot 4.10: Generated Timetable Page

4.5 DATABASE AND SECURITY

The Database & Security module manages the storage of data and ensures its protection from unauthorized access or breaches.

Data Storage: The system's database securely stores all relevant data, including user information, course details, timetable schedules, and preferences. Data integrity and consistency are maintained to ensure the accuracy and reliability of information.



```
CREATE DATABASE CollegeTimetable;
USE CollegeTimetable;

CREATE TABLE Users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    userId VARCHAR(50) UNIQUE,
    password VARCHAR(50),
    role ENUM('student', 'teacher','admin')
);

ALTER TABLE Users
ADD COLUMN isFirstLogin BOOLEAN DEFAULT TRUE;

 CREATE TABLE courses (
    name VARCHAR(255) PRIMARY KEY,
    credits INT NOT NULL
);

CREATE TABLE rooms (
    name VARCHAR(100) PRIMARY KEY
```

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

```
#  Time      Action                                                   Message               Duration / Fetch
1  18:42:20  SELECT * FROM collegetimetable.setuppractical LIMIT 0, 1000  24 row(s) returned   0.016 sec / 0.000 sec
2  18:42:33  SELECT * FROM collegetimetable.setupsubject LIMIT 0, 1000    30 row(s) returned   0.032 sec / 0.000 sec
```

Connection Details

Name:	Local instance MySQL80
Host:	localhost
Port:	3306
Login User:	root
Current User:	root@localhost
SSL cipher:	TLS_AES_128_GCM_SHA256

Server

Product:	MySQL Community Server - GPL
Version:	8.0.39

Screenshot 4.11: Database Storage

CHAPTER 5

RESULT

AND

DISCUSSION

RESULT AND DISCUSSION

The web application developed successfully automates the generation of academic timetables by utilizing Java Servlets and MySQL as the back-end. The application successfully schedules classes, teachers, and classrooms through a user-friendly and interactive interface. Once executed, the system enables administrators to securely log in and view scheduling options according to their profiles. The smart scheduling algorithm, augmented by a genetic algorithm, effectively addresses several constraints like faculty availability, course timings, and room assignment. The genetic algorithm applies selection, crossover, and mutation operations to resolve clashes and optimize timetabling generation.

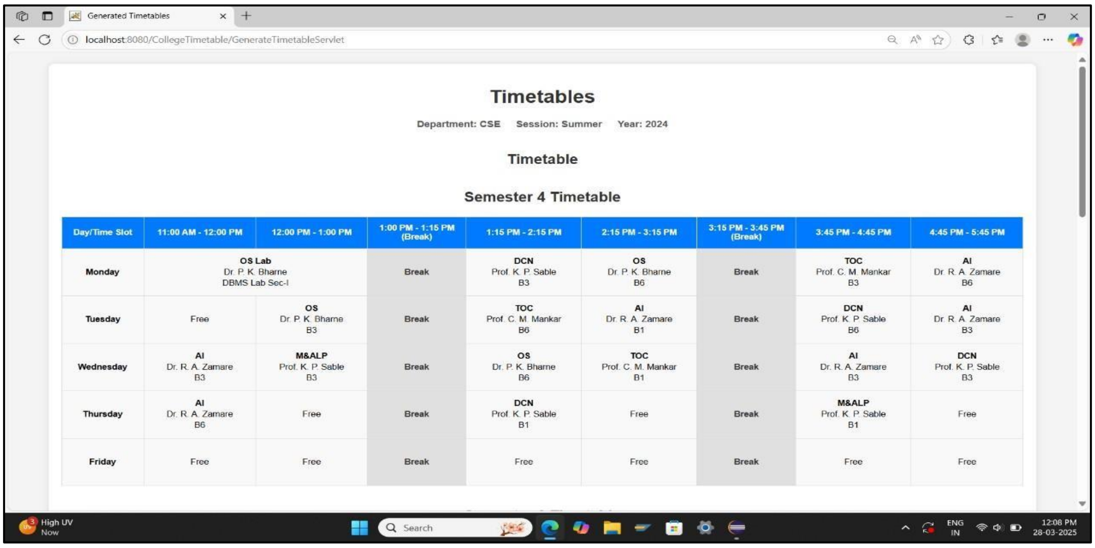
To evaluate the effectiveness of different algorithmic strategies for scheduling, a comparative analysis is shown below:

Table No. 5.1: Algorithm Comparison

Algorithm	Approach	Efficiency	Flexibility	Scalability
Brute Force	Tries all possible solutions	Extremely slow	Not adaptable	Not scalable
Greedy Algorithm	Makes local best choices	Fast but suboptimal	Limited flexibility	Struggles with complexity
Backtracking	Recursive solution search	Slower for large problems	May get stuck	Limited scalability
Genetic Algorithm	Evolutionary optimization	Efficient & near-optimal	Adapts to constraints	Highly scalable

After the generation of the schedule, users are presented with a "Generated Timetable" on the interface, providing a clear visual overview of the classes allocated across different slots. Prior to publishing, a request permission prompt "Requesting Permission to Publish Timetable" is shown to the administrator, providing controlled publishing of the schedule. On successful approval, the system displays a "Message Showing Success of Generated Timetable" to ensure that the timetable has been saved successfully in the backend and is viewable by all stakeholders according to their role-based permissions.

The performance of the application was observed to be reliable and scalable, handling real-time updates and ensuring conflict-free scheduling. This approach significantly reduces the manual workload and chances of human error, providing a structured and efficient solution for academic scheduling needs.



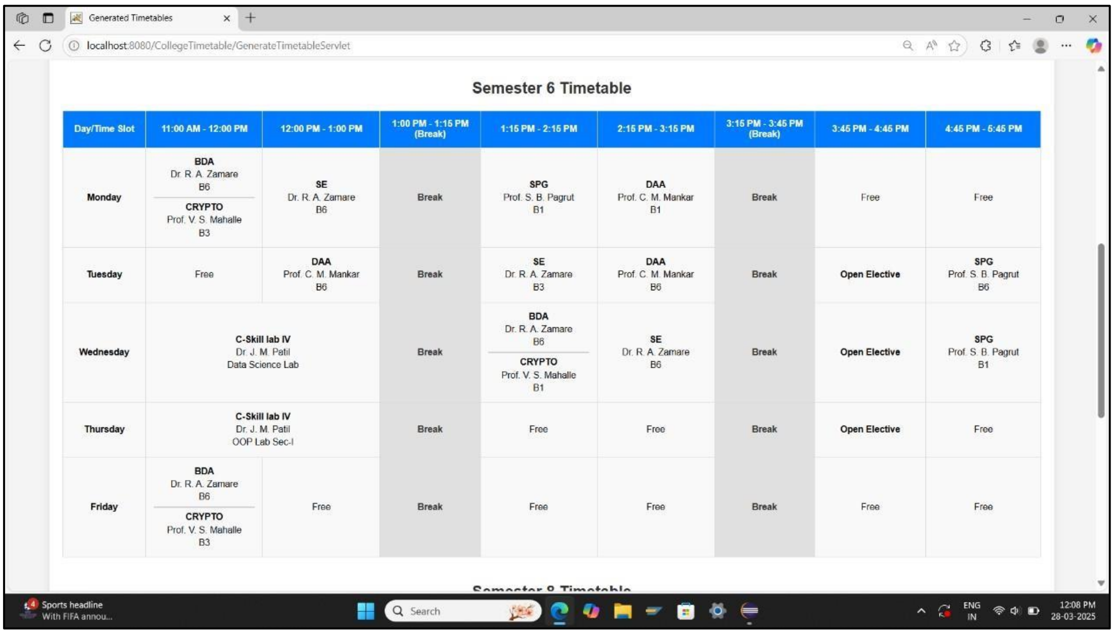
Timetables

Department: CSE Session: Summer Year: 2024

Timetable

Semester 4 Timetable

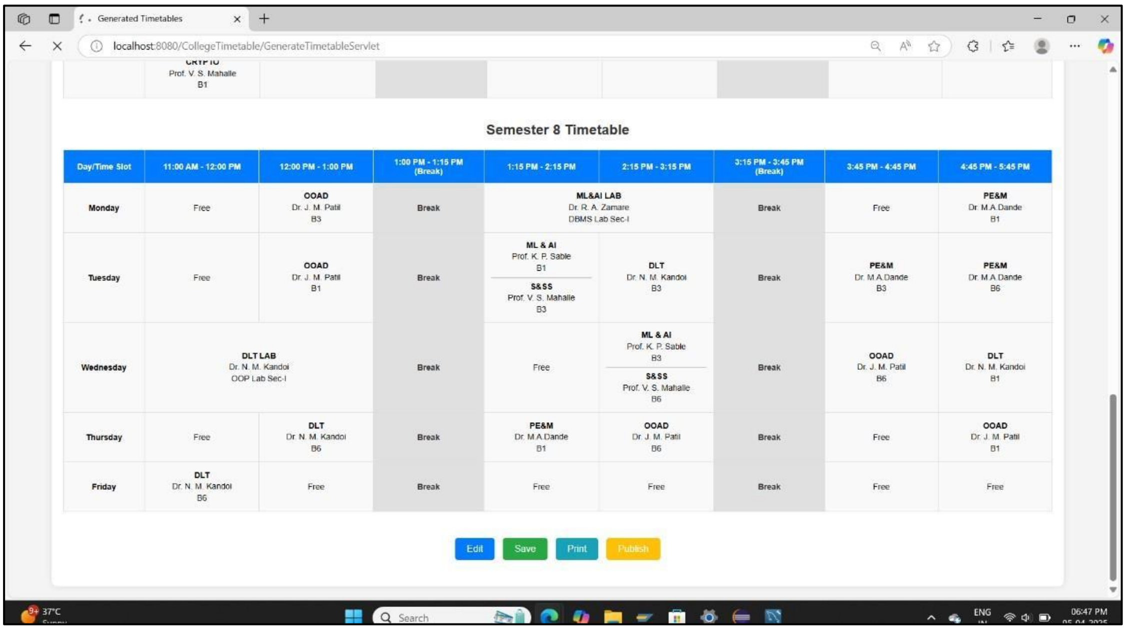
Day/Time Slot	11:00 AM - 12:00 PM	12:00 PM - 1:00 PM	1:00 PM - 1:15 PM (Break)	1:15 PM - 2:15 PM	2:15 PM - 3:15 PM	3:15 PM - 3:45 PM (Break)	3:45 PM - 4:45 PM	4:45 PM - 5:45 PM
Monday	**OS Lab** Dr. P. K. Bharne DBMS Lab Sec-I		Break	**DCN** Prof. K. P. Sable B3	**OS** Dr. P. K. Bharne B6	Break	**TOC** Prof. C. M. Mankar B3	**AI** Dr. R. A. Zamare B6
Tuesday	Free	**OS** Dr. P. K. Bharne B3	Break	**TOC** Prof. C. M. Mankar B6	**AI** Dr. R. A. Zamare B1	Break	**DCN** Prof. K. P. Sable B6	**AI** Dr. R. A. Zamare B3
Wednesday	**AI** Dr. R. A. Zamare B3	**M&ALP** Prof. K. P. Sable B3	Break	**OS** Dr. P. K. Bharne B6	**TOC** Prof. C. M. Mankar B1	Break	**AI** Dr. R. A. Zamare B3	**DCN** Prof. K. P. Sable B3
Thursday	**AI** Dr. R. A. Zamare B6	Free	Break	**DCN** Prof. K. P. Sable B1	Free	Break	**M&ALP** Prof. K. P. Sable B1	Free
Friday	Free	Free	Break	Free	Free	Break	Free	Free



Semester 6 Timetable

Day/Time Slot	11:00 AM - 12:00 PM	12:00 PM - 1:00 PM	1:00 PM - 1:15 PM (Break)	1:15 PM - 2:15 PM	2:15 PM - 3:15 PM	3:15 PM - 3:45 PM (Break)	3:45 PM - 4:45 PM	4:45 PM - 5:45 PM
Monday	**BDA** Dr. R. A. Zamare B6 **CRYPTO** Prof. V. S. Mahalle B3	**SE** Dr. R. A. Zamare B6	Break	**SPG** Prof. S. B. Pagrut B1	**DAA** Prof. C. M. Mankar B1	Break	Free	Free
Tuesday	Free	**DAA** Prof. C. M. Mankar B6	Break	**SE** Dr. R. A. Zamare B3	**DAA** Prof. C. M. Mankar B6	Break	**Open Elective**	**SPG** Prof. S. B. Pagrut B6
Wednesday	**C-Skill lab IV** Dr. J. M. Patil Data Science Lab		Break	**BDA** Dr. R. A. Zamare B6 **CRYPTO** Prof. V. S. Mahalle B1	**SE** Dr. R. A. Zamare B6	Break	**Open Elective**	**SPG** Prof. S. B. Pagrut B1
Thursday	**C-Skill lab IV** Dr. J. M. Patil OOP Lab Sec-I		Break	Free	Free	Break	**Open Elective**	Free
Friday	**BDA** Dr. R. A. Zamare B6 **CRYPTO** Prof. V. S. Mahalle B3	Free	Break	Free	Free	Break	Free	Free

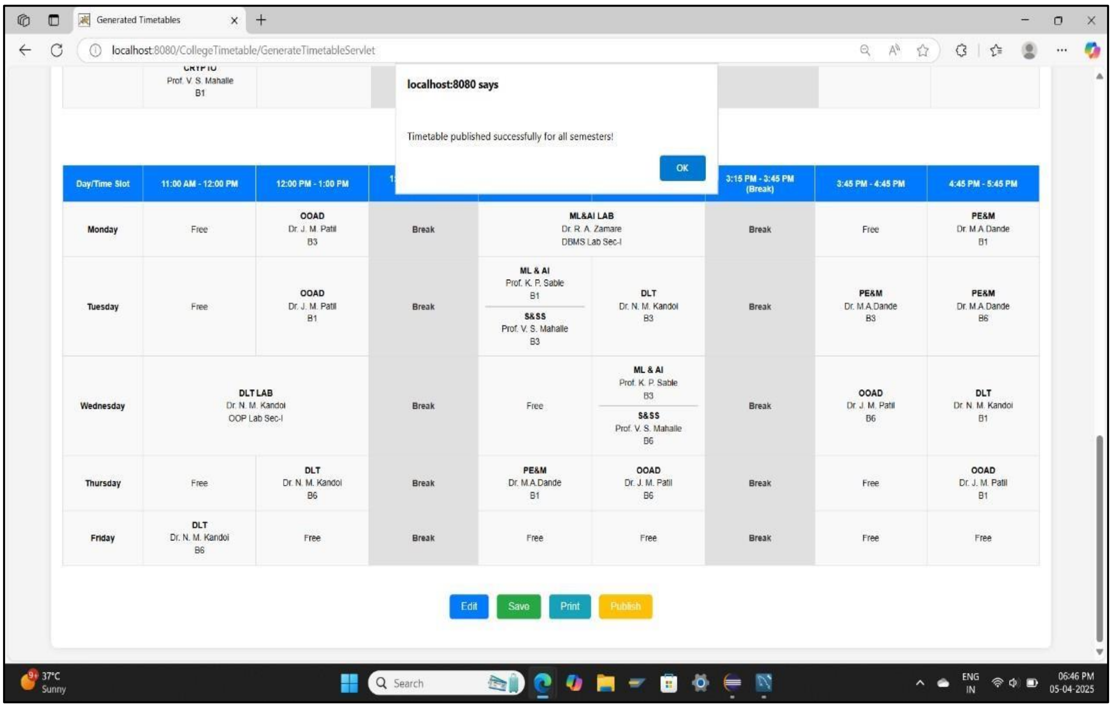
Semester 8 Timetable



Semester 8 Timetable

Day/Time Slot	11:00 AM - 12:00 PM	12:00 PM - 1:00 PM	1:00 PM - 1:15 PM (Break)	1:15 PM - 2:15 PM	2:15 PM - 3:15 PM	3:15 PM - 3:45 PM (Break)	3:45 PM - 4:45 PM	4:45 PM - 5:45 PM
Monday	Free	OOAD Dr. J. M. Patil B3	Break	ML&AI LAB Dr. R. A. Zamare DBMS Lab Sec-I		Break	Free	PE&M Dr. M.A.Dande B1
Tuesday	Free	OOAD Dr. J. M. Patil B1	Break	ML & AI Prof. K. P. Sable B1 S&SS Prof. V. S. Mahalle B3	DLT Dr. N. M. Kandoi B3	Break	PE&M Dr. M.A.Dande B3	PE&M Dr. M.A.Dande B6
Wednesday	DLT LAB Dr. N. M. Kandoi OOP Lab Sec-I		Break	Free	ML & AI Prof. K. P. Sable B3 S&SS Prof. V. S. Mahalle B6	Break	OOAD Dr. J. M. Patil B6	DLT Dr. N. M. Kandoi B1
Thursday	Free	DLT Dr. N. M. Kandoi B6	Break	PE&M Dr. M.A.Dande B1	OOAD Dr. J. M. Patil B6	Break	Free	OOAD Dr. J. M. Patil B1
Friday	DLT Dr. N. M. Kandoi B6	Free	Break	Free	Free	Break	Free	Free

Screenshot 5.1.1: Generated Timetable



localhost:8080 says

Timetable published successfully for all semesters!

Day/Time Slot	11:00 AM - 12:00 PM	12:00 PM - 1:00 PM	1			3:15 PM - 3:45 PM (Break)	3:45 PM - 4:45 PM	4:45 PM - 5:45 PM
Monday	Free	OOAD Dr. J. M. Patil B3	Break	ML&AI LAB Dr. R. A. Zamare DBMS Lab Sec-I		Break	Free	PE&M Dr. M.A.Dande B1
Tuesday	Free	OOAD Dr. J. M. Patil B1	Break	ML & AI Prof. K. P. Sable B1 S&SS Prof. V. S. Mahalle B3	DLT Dr. N. M. Kandoi B3	Break	PE&M Dr. M.A.Dande B3	PE&M Dr. M.A.Dande B6
Wednesday	DLT LAB Dr. N. M. Kandoi OOP Lab Sec-I		Break	Free	ML & AI Prof. K. P. Sable B3 S&SS Prof. V. S. Mahalle B6	Break	OOAD Dr. J. M. Patil B6	DLT Dr. N. M. Kandoi B1
Thursday	Free	DLT Dr. N. M. Kandoi B6	Break	PE&M Dr. M.A.Dande B1	OOAD Dr. J. M. Patil B6	Break	Free	OOAD Dr. J. M. Patil B1
Friday	DLT Dr. N. M. Kandoi B6	Free	Break	Free	Free	Break	Free	Free

Screenshot 5.1.2: Message Showing Success of Publishing Timetable

CHAPTER 6

CONCLUSION

CONCLUSION

6.1 CONCLUSION

Our system addresses the challenges of managing faculty members and scheduling lectures effectively. Through automation and advanced algorithms, it streamlines the process of generating timetables for various courses and semesters. With its user-friendly interface and efficient processing, it significantly reduces the time and effort required for timetable creation. By minimizing manual intervention and errors, our system optimizes productivity and resource utilization, ultimately saving valuable time and manpower.

6.2 CONTRIBUTIONS

The implementation of automation and advanced algorithms in our system marks a significant advancement in efficiently managing faculty members and scheduling lectures. Through the utilization of these technologies, we have successfully streamlined the cumbersome process of timetable generation across various courses and semesters. A standout feature of our system is its user-friendly interface, designed to ensure accessibility even for users with limited technical expertise. This accessibility not only enhances user experience but also encourages widespread adoption among faculty members and administrative staff. Moreover, the system's efficient processing capabilities play a crucial role in reducing the time and effort traditionally associated with timetable creation. By automating repetitive tasks and optimizing resource allocation, we have significantly minimized manual intervention and errors, resulting in more accurate timetables and reducing the need for constant adjustments.

6.3SCOPE FOR FUTURE WORK

The developed timetable generation system provides a solid foundation for automated academic scheduling. However, several enhancements can be incorporated to extend its utility and adaptability. In the future, the system can be expanded to support multi-campus scheduling, allowing centralized control over larger academic institutions. Integration of AI and machine learning techniques can further refine the genetic algorithm to learn from past scheduling patterns and improve decision-making over time. Additionally, incorporating a mobile-friendly responsive design or a dedicated mobile application would ensure greater accessibility for students and faculty. Features like automated conflict notifications via email or SMS, drag-and-drop timetable customization, and analytics dashboards for resource utilization can significantly enhance user engagement and administrative efficiency. Lastly, integration with academic ERP systems and calendar apps like Google Calendar can provide a seamless and synchronized academic planning experience.

REFERENCES

REFERENCES

[1] A. H. Khan and T. Imtiaz, "A Novel Genetic Algorithm-Based Timetable Generator for Optimized University Timetable Solution," 2024 International Conference on Engineering & Computing Technologies (ICECT), pp. 1-6, 2024. DOI: 10.1109/ICECT61618.2024.10581296.

[2] G. Maneesha, T. Deepika, S. BhanuSri, N. RaviKumar, and P. Siva Nagamani, "Automatic Time Table Generation Using Genetic Algorithm," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 8, no. 7, pp. 1-7, July 2021.

[3] V. S, D. S, B. G. L. S, S. V, V. S, and B. Gowda, "Design and Implementation of Time Table Generator," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 9, no. 6, pp. 1-5, June 2022.

[4] R. Firke, P. Bhabad, O. Gangarde, A. Magar, and A. Tawlare, "Automatic Timetable Generation System," International Journal of Creative Research Thoughts (IJCRT), vol. 11, no. 10, pp. 1-6, Oct. 2023.

[5] S. Thakare, T. Nikam, and M. Patil, "Automated Timetable Generation Using Genetic Algorithm," International Journal of Engineering Research & Technology (IJERT), vol. 9, no. 7, pp. 1-5, July 2020.

[6] S. S. A. Ahmed, A. A. Ansari, and A. Z. Ansari, "Automated Timetable Generator," International Journal for Innovative Research in Science & Technology (IJIRST), vol. 1, no. 11, pp. 1-5, Apr. 2015.

[7] S. Nagtilak, N. Dongare, A. Salave, P. Dongare, G. Yeole, and P. Satarkar, "Smart Timetable System Using Machine Learning and Artificial Intelligence," International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), vol. 3, no. 5, pp. 1-6, June 2023.

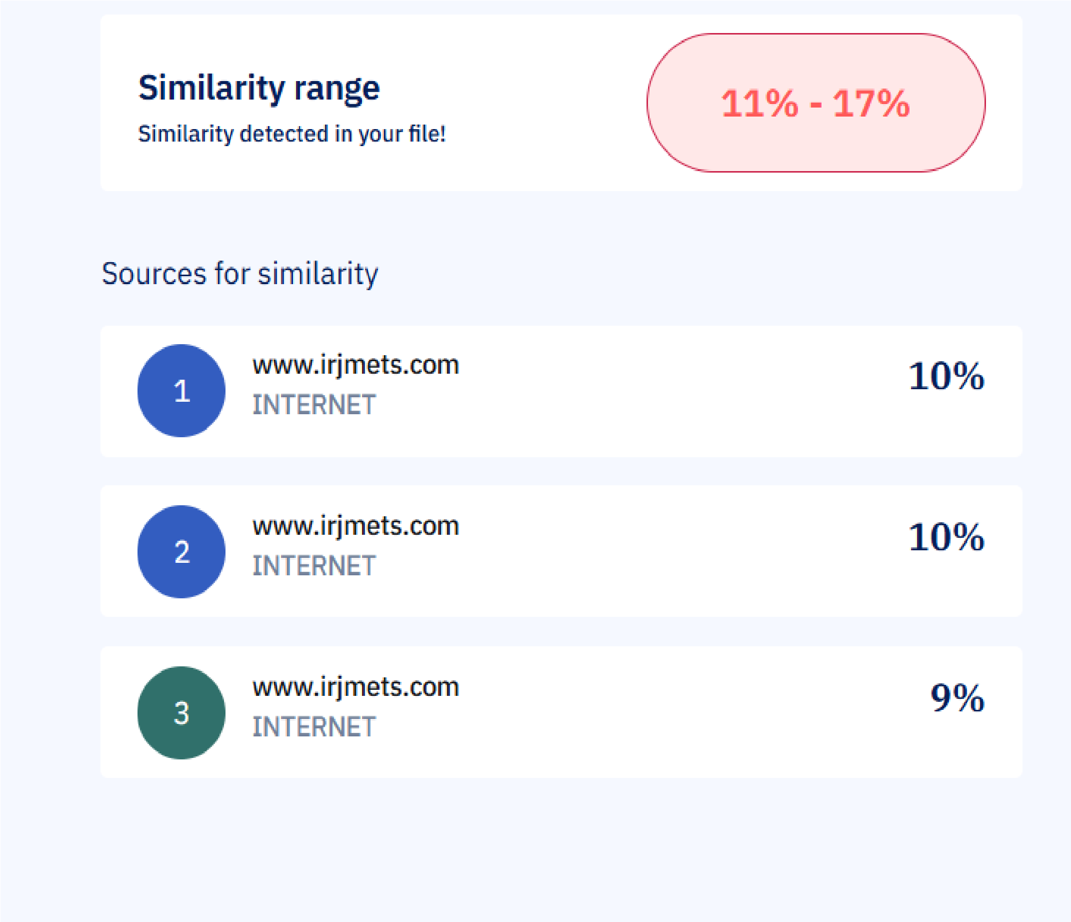
[8] P. P. Rathod, K. K. Lodhiya, M. P., P. S. Student CSE, and S. C., "Automatic Timetable Generator," International Journal of Research in Science & Engineering, 2016.

[9] Deeksha C. S., A. Kavya Reddy, A. Nagambika, A. Castelino, and K. Panimozhi, "Automatic Timetable Generation System," 2015.

[10] A. Chowdhary, S. D. Priyanka Kakde, and D. G. Rupal Rushiya, "Timetable Generation System," International Journal of Computer Science and Mobile Computing, Feb. 2015.

DISSEMINATION OF WORK

PLAGIARISM REPORT



Similarity range

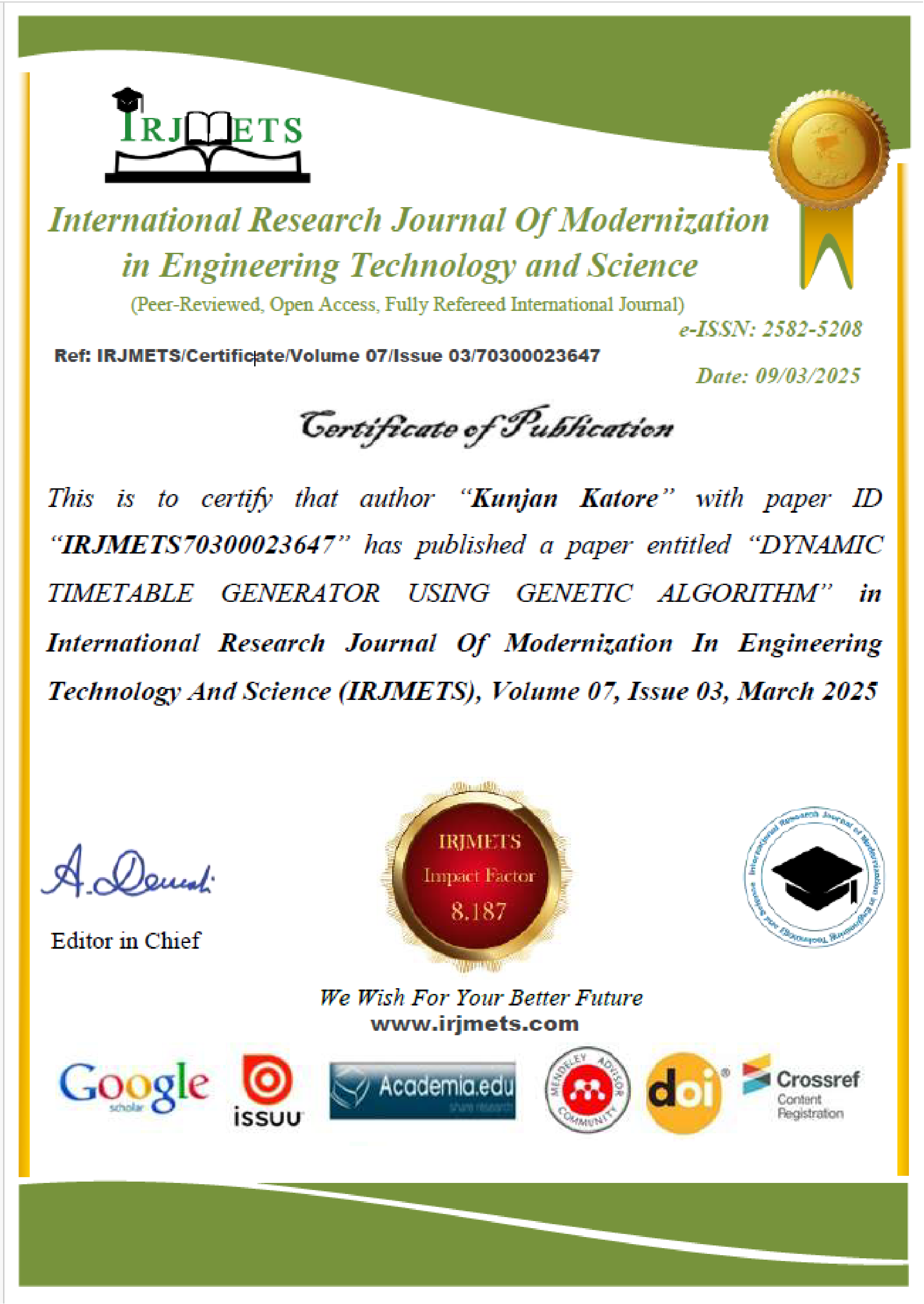
Similarity detected in your file!

11% - 17%

Sources for similarity

1	www.irjmets.com INTERNET	10%
2	www.irjmets.com INTERNET	10%
3	www.irjmets.com INTERNET	9%

CERTIFICATES OF PUBLISHED PAPER



International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 07/Issue 03/70300023647

Date: 09/03/2025

Certificate of Publication

*This is to certify that author "**Kunjan Katore**" with paper ID "**IRJMETS70300023647**" has published a paper entitled "DYNAMIC TIMETABLE GENERATOR USING GENETIC ALGORITHM" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 07, Issue 03, March 2025***

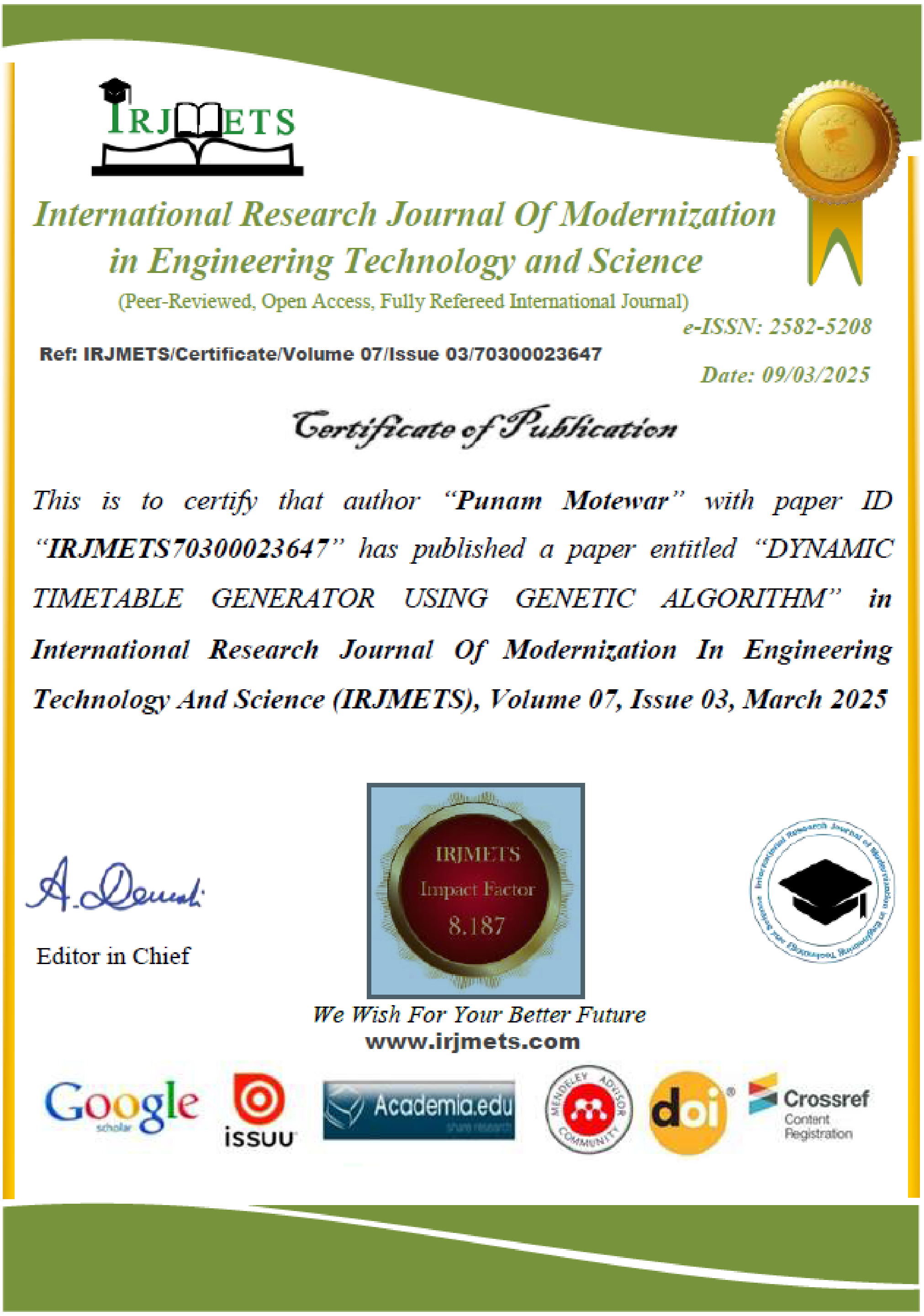
Editor in Chief

IRJMETS Impact Factor 8.187

We Wish For Your Better Future

www.irjmets.com

Google scholar
issuu
Academia.edu share research
Mendeley Advisor Community
doi
Crossref Content Registration



IRJMETS

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 07/Issue 03/70300023647

Date: 09/03/2025

Certificate of Publication

*This is to certify that author "**Punam Motewar**" with paper ID "**IRJMETS70300023647**" has published a paper entitled "DYNAMIC TIMETABLE GENERATOR USING GENETIC ALGORITHM" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 07, Issue 03, March 2025***

Editor in Chief

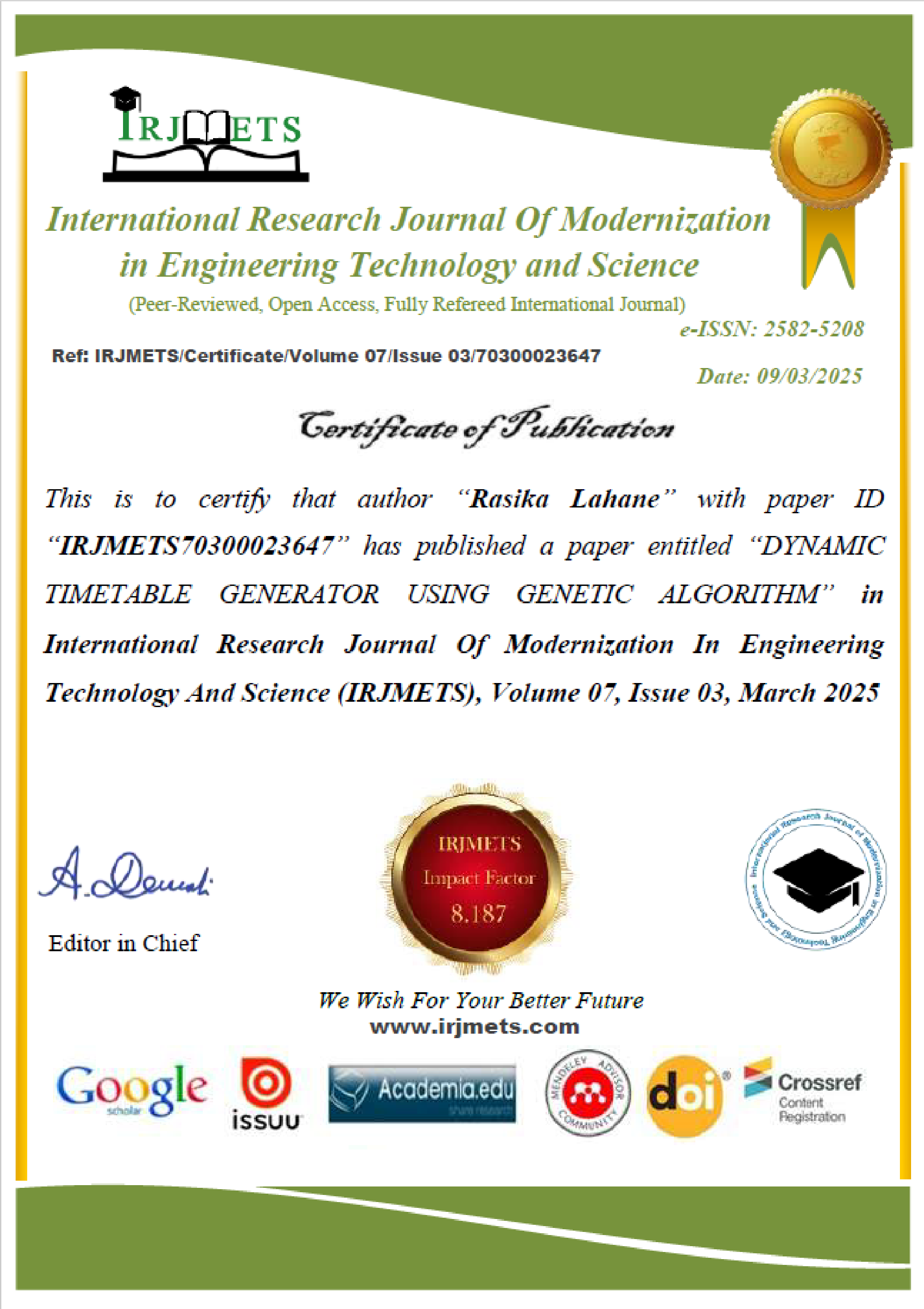
IRJMETS Impact Factor 8.187

International Research Journal of Modernization in Engineering Technology and Science

We Wish For Your Better Future

www.irjmets.com

Google scholar issuu Academia.edu share research Mendeley Advisor Community doi Crossref Content Registration



IRJMETS

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 07/Issue 03/70300023647

Date: 09/03/2025

Certificate of Publication

*This is to certify that author “**Rasika Lahane**” with paper ID “**IRJMETS70300023647**” has published a paper entitled “DYNAMIC TIMETABLE GENERATOR USING GENETIC ALGORITHM” **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 07, Issue 03, March 2025***

Editor in Chief

IRJMETS
Impact Factor
8.187

We Wish For Your Better Future

www.irjmets.com

Google scholar

issuu

Academia.edu

doi

Crossref
Content
Registration

IRJMETS

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 07/Issue 03/70300023647

Date: 09/03/2025

Certificate of Publication

*This is to certify that author “**Chaitali Nakhate**” with paper ID “**IRJMETS70300023647**” has published a paper entitled “DYNAMIC TIMETABLE GENERATOR USING GENETIC ALGORITHM” **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 07, Issue 03, March 2025***

Editor in Chief

IRJMETS
Impact Factor
8.187

We Wish For Your Better Future
www.irjmets.com

PROJECT GROUP DETAILS

	Name: **Kunjan P. Katore** **Address:** Indira Gandhi Ward, Hinganghat **Mobile No:** 7058644046 **Email ID:** kunjankatore@gmail.com
	Name: **Punam R. Motewar** **Address:** Karla Road, Wardha **Mobile No:** 9307153047 **Email ID:** punammotewar@gmail.com
	Name: **Rasika D. Lahane** **Address:** Civil line Paratwada **Mobile No:** 9022337590 **Email ID:** rasikalahane2@gmail.com
	Name: **Chaitali S. Nakhate** **Address:** Indira Gandhi Ward, Hinganghat **Mobile No:** 9022860737 **Email ID:** chaitalinakhate20@gmail.com